U.S. DEPARTMENT OF COMMERCE
National Technical Information Service

AD-A027 530

DISPERSION CONTROL SYSTEM FOR SOUNDING ROCKETS

AIR FORCE GEOPHYSICS LABORATORY

4 MARCH 1976

AFGL-TR-76-0044
INSTRUMENTATION PAPERS, NO. 246

Dispersion Control System for Sounding Rockets

JAMES R. PICKELL, Capt, USAF

4 March 1976

Approved for public release; distribution unlimited.

AEROSPACE INSTRUMENTATION DIVISION PROJECT 7659
AIR FORCE GEOPHYSICS LABORATORY
HANSCOM AFB, MASSACHUSETTS 01731

AIR FORCE SYSTEMS COMMAND, USAF

This technical report has been reviewed and is approved for publication.

FOR THE COMMANDER:

John Howard

Chief Scientist

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER AFGL-TR-76-0044	2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) DISPERSION CONTROL SYSTEM FOR SOUNDING ROCKETS		5. TYPE OF REPORT & PERIOD COVERED Scientific. Interim. 6. PERFORMING ORG. REPORT NUMBER IP No. 246
7. AUTHOR(s) James R. Pickell, Capt, USAF		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Air Force Geophysics Laboratories .CR) Hanscom AFB Massachusetts 01731		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 62101F 76590401
11. CONTROLLING OFFICE NAME AND ADDRESS Air Force Geophysics Laboratories (LCR) Hanscom AFB Massachusetts 01731		12. REPORT DATE [illegible] March 1976 13. NUMBER OF PAGES 79
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release, distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Control sounding rocket dispersion

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

High altitude sounding rockets have always presented a problem to small test ranges such as WSMR because of rocket dispersion. This report presents a strap-on dispersion control system employing a 16-bit microprocessor as its brains that requires only software changes to accommodate different sounding rocket vehicles. Mostly the report contains a detailed explanation of the electronic hardware used, and the software developed, for a Paiute-Tomahawk sounding rocket.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

Unclassified

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

Contents

Preceding page blank

Contents

Illustrations

Dispersion Control System for Sounding Rockets

1. INTRODUCTION

High altitude sounding rockets have always presented a problem to small test ranges such as WSMR because of rocket dispersion. This report presents a strap-on dispersion control system employing a 16-bit microprocessor as its brains, that requires only software changes to accommodate different sounding rocket vehicles.

The Paiute-Tomahawk sounding rocket was chosen as the first test vehicle for this system because it is the most difficult to control. Figure 1 shows the payload as it was hung by ropes from the launcher during TM checks at WSMR. The actuator is a pneumatic position, proportional canard control manufactured by Chandler Evans Inc. A 6000 psi helium gas bottle, visible in the illustration, provides the energy to move the fins. An exercise port is available on the actuator for connection of an external helium source, providing a means of exercising the fins without firing a pyrotechnic which opens the helium gas bottle valve. Figure 2 shows the rack in the control section. A modified MIDAS platform (Gyro) manufactured by the Space Vector Corporation provided the error signals for the control system. Note that the removable eyelets at the top of the rack allowed for [illegible] insertion and removal from the control housing section. Figure 3 shows the control system electronics mounted on four wire wrap boards with the 16-bit microprocessor mounted on the wall of the electronics box for heat sinking purposes. The wire

(Received for publication 3 March 1976)

Preceding page blank

wrap boards are manufactured by Mupac Corp. and used because of their unique 108 pin socket connector. The 16-bit microprocessor is the TDY-52B manufactured by Teledyne Systems Company and discussed in greater detail in Section 2. 3.

Operation of the control system was only during the coast phase of flight, that is, T+5 sec to T+17 sec with second stage ignition at T+20 sec. The system is capable of guiding through second stage, however, Range Safety required early shut down so they could observe the gyro output and determine if it is safe to enable the second stage. The system was launched 16 October 1975 at WSMR with a radar impact within a mile of a computer simulated impact.

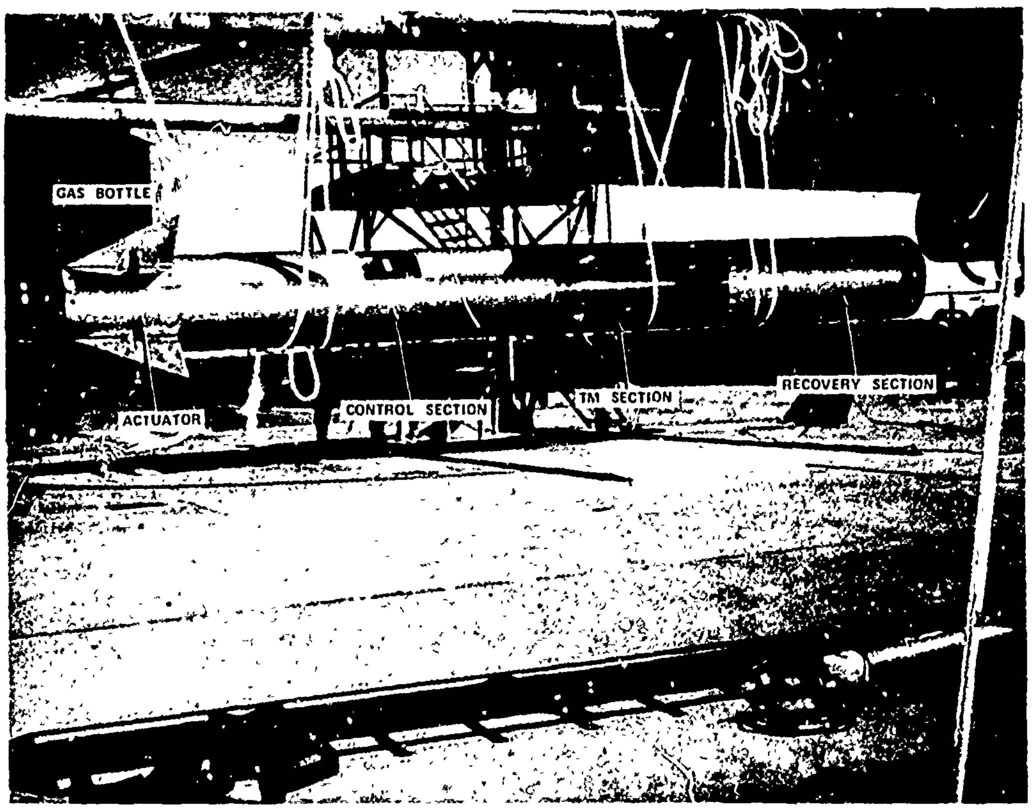

Figure 1. Dispersion Control 9-in. Dia Payload

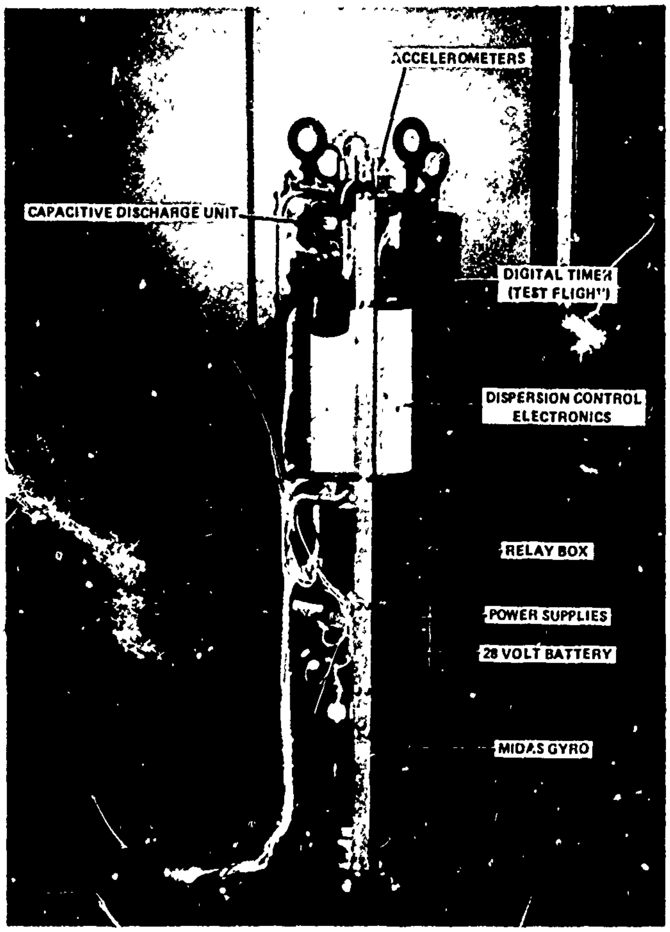

Figure 2. Dispersion Control Rack

2. DISPERSION CONTROL ELECTRONICS

A detailed discussion of the important electronic circuitry used by the Dispersion Control Electronics is described in the following sections. The complete circuit diagram is not contained in this report because of its size.

2.1 Block Diagram

Figure 4 is a simplified block diagram of the dispersion control system electronics. The heart of the system is a 16-bit microprocessor, TDY-52B, requiring only a 512 word by 16-bit memory to perform all control algorithms.

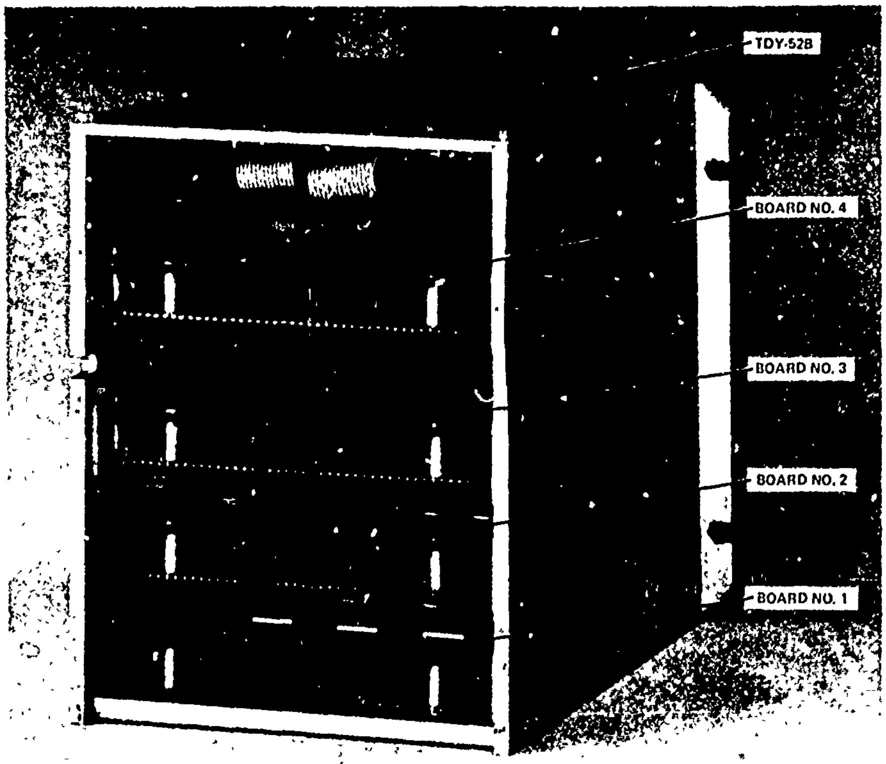

Figure 3. Dispersion Control Electronics

At rocket liftoff a G-switch activates the Interrupt Start which delivers a clock pulse approximately every 10 ms to the TDY-52B interrupt line. Every 10 ms the TDY-52B will increment a memory location, that is, the TDY-52B will also be the system clock. Five sec after liftoff the TDY-52B will pulse the Initiate Actuator signal releasing gas into the chamber of the pneumatic actuator. Until 17 sec after liftoff when the Lock Fins signal is pulsed the TDY-52B will sample the Gyro outputs every 10 ms and provide an output to the canards via the Pitch & Yaw Command A&B signals. At 20 sec the 2nd State Initiate line will be pulsed and the TDY-52B will halt.

Those blocks which require detailed explanation are discussed in the following sections.

2.2 Gyro Resolver

The Gyro Resolver transforms the gyro roll, pitch and yaw encoder outputs into positional information.

Figure 5 is the basic circuit used to transform the gyro encoder outputs, signals A and B, into positional information, Up/Down Binary counter outputs.

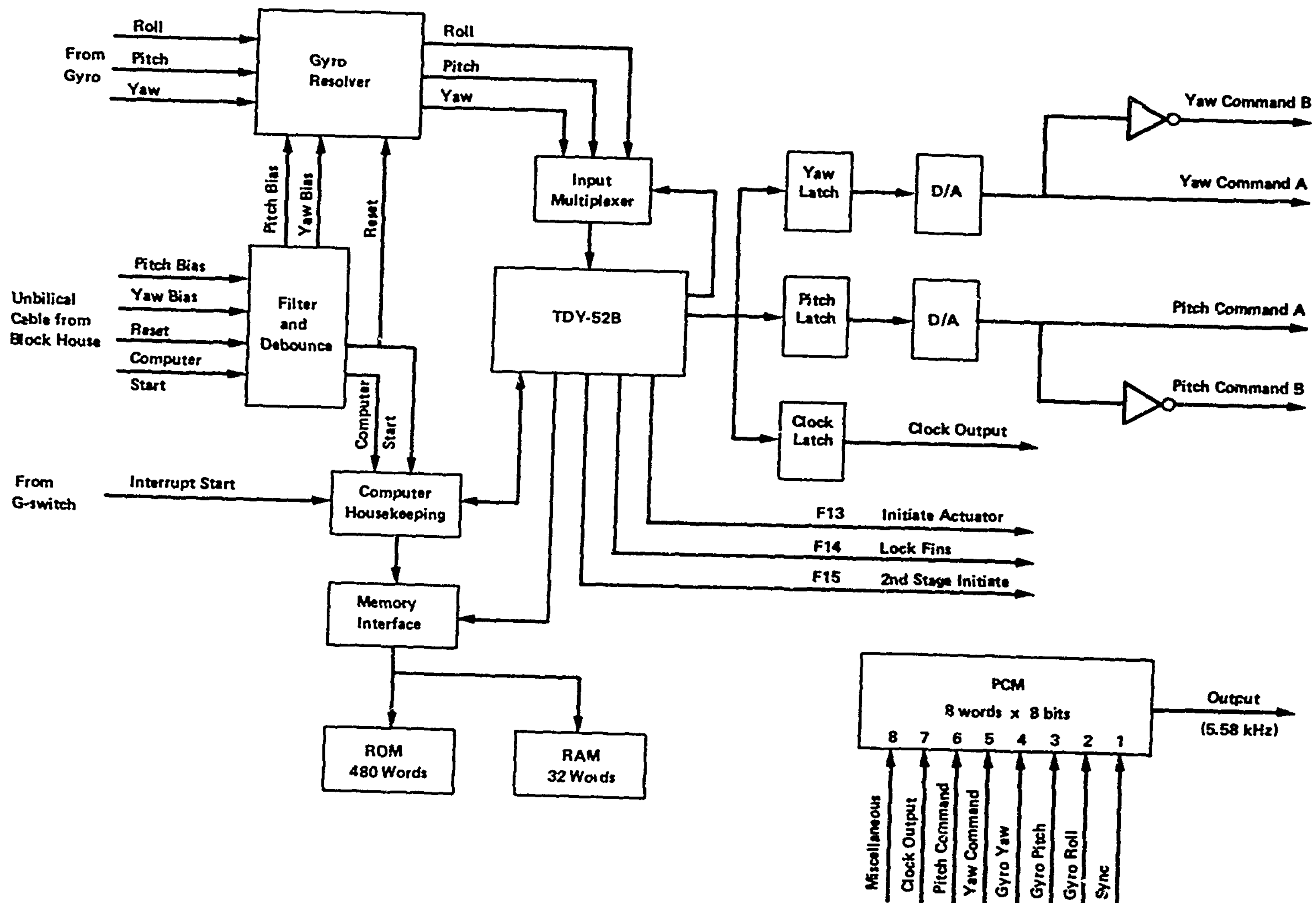

Figure 4. Dispersion Control Simplified Block Diagram

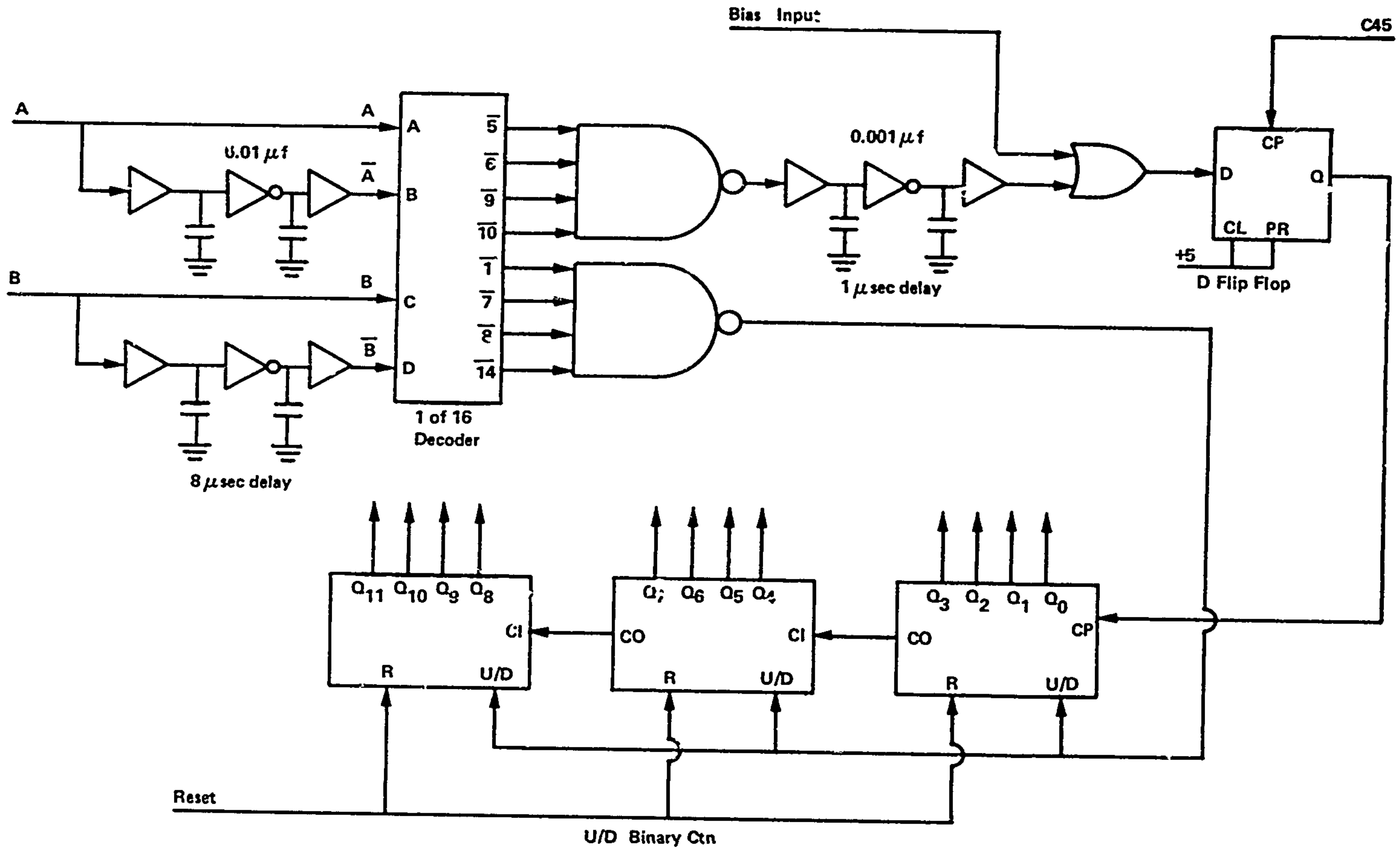

Figure 5. Basic Gyro Resolver Circuit

There are three such circuits used by the Gyro Resolver, one each for Roll, Pitch and Yaw. Signals A and B, Figure 6, indicate each .08789 degree change in gyro position and the direction of change, positive or negative from the previous position. Thus each change of state between A and B is a change in gyro position of .08789 degrees and the knowledge of the previous A and B state immediately following a change of state indicates whether the gyro moved in a positive or negative direction.

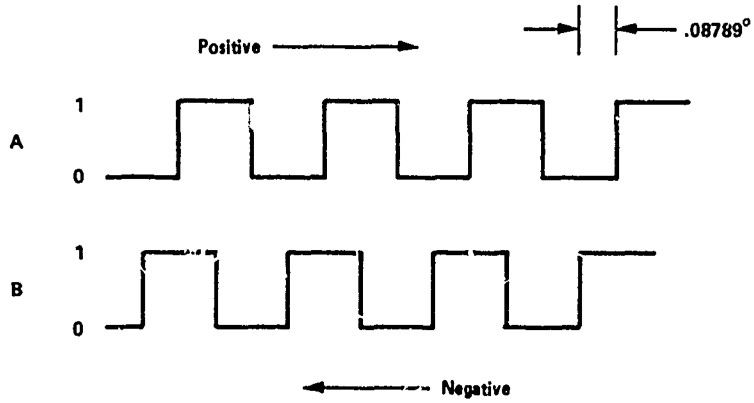

Figure 6. Gyro Encoder Outputs

The truth table in Figure 7 indicates all possible input states to the 1 of 16 decoder and the resultant outputs to the binary counters. The truth table is read from top to bottom for each direction positive or negative. To change direction jump between identical input stable states and continue to read down.

Ignore for the time being the OR gate and D Flip Flop in Figure 5. The 8 μsec delays allow the transition states to exist long enough to provide a clock pulse, CP, whose positive going edge will increment or decrement the binary counters as determined by the Up/down, U/D, signal. The 1 μsec delay insures the stability of the U/D signal before the CP signal reaches the counters. Synchronization of the Gyro resolver with the TDY-52B is accomplished by the D Flip Flop which is clocked by C45, a clock signal generated by the TDY-52B. Presetting of the binary counters is accomplished serially through the OR gate using the Bias Input.

CONDITION	INPUT				OUTPUT		
	$\bar{B}$	B	$\bar{A}$	A	U/D	CP	
Stable	1	0	1	0	0	0	
Transition	1	0	1	1	0	1	
Stable	1	0	0	1	0	0	
Transition	1	1	0	1	0	1	Decrement (Negative)
Stable	0	1	0	1	0	0	Counters
Transition	0	1	0	0	0	1	
Stable	0	1	1	0	0	0	
Transition	0	0	1	0	0	1	
Stable	1	0	1	0	0	0	
Transition	1	1	1	0	1	1	
Stable	0	1	1	0	0	0	
Transition	0	1	1	1	1	1	
Stable	0	1	0	1	0	0	Increment (Positive)
Transition	0	0	0	1	1	1	Counters
Stable	1	0	0	1	0	0	
Transition	1	0	0	0	1	1	
Stable	1	0	1	0	0	0	

Figure 7. Gyro Resolver Truth Table

2.3 TDY-52B Microcomputer

The TDY-52B is a 16-bit parallel processor packaged in a 2 in. x 2 in. x 2 in. (Figure 12) hermetically-sealed module, dimensions are exclusive of its 120 pins. Teledyne Systems Company is the manufactor of this hybrid computer based on National Semiconductors IMP-16C micro-computer. The TDY 52B has the following features:

Word Length	16 Bits
Instruction Set	60 (implemented by CPU resident microprogram)
Arithmetic	Parallel, binary, fixed point, two's complement Multiply, Divide, Double precision Add and Subtract
Memory	Must be provided externally 16 word Last-In/First Out Stack Internal
Addressing	Page Size of 256 Words. For direct and indirect modes: Absolute Relative to Program Counter Relative to Accumulator 2 (indexed) Relative to Accumulator 3 (indexed)
Accumulators	4

Input/Output and Control	16 bit data - memory input port
	16 bit data - peripheral input port
	16 bit data - output bus
	16 bit address bus
	6 general-purpose output flags
	4 general-purpose jump-condition inputs
	1 general interrupt input
	1 control panel interrupt input

Figure 8 shows a simplified block diagram of the TDY 52B. The CPU is the heart of the TDY 52B which is configured around MOS/LSI devices, as shown in Figure 9. The MOS/LSI devices consists of two CROM's (Control Read Only Memory) and four RALU's (Register and Arithmetic Logic Units). Each RALU handles 4 bits, and a 16 bit CPU is formed by connecting four RALU's in parallel. A 4-bit wide control bus is used by the CROM's to communicate most of the control information to the RALU's

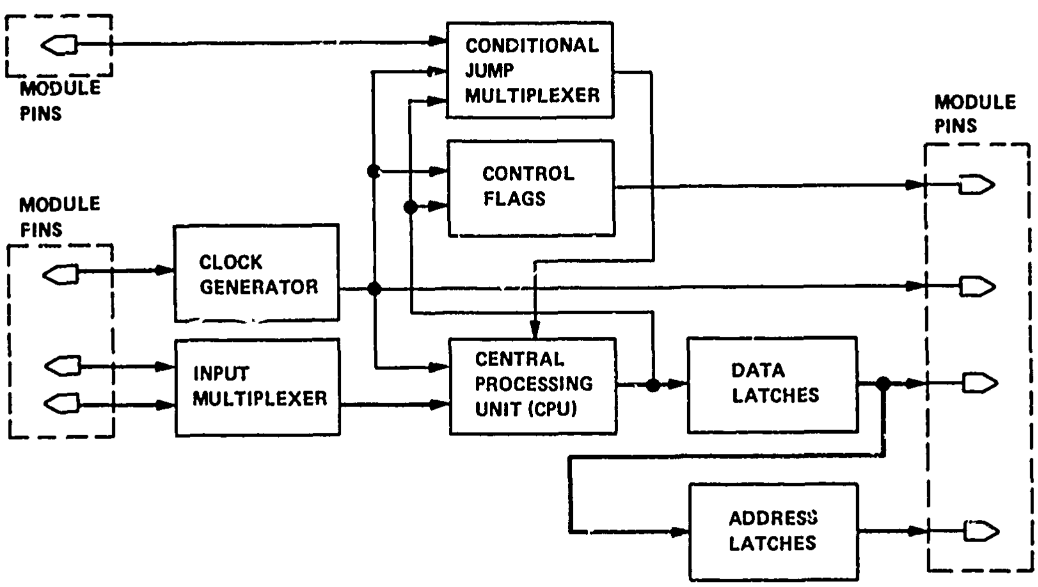

Figure 8. TDY-52B Simplified Block Diagram

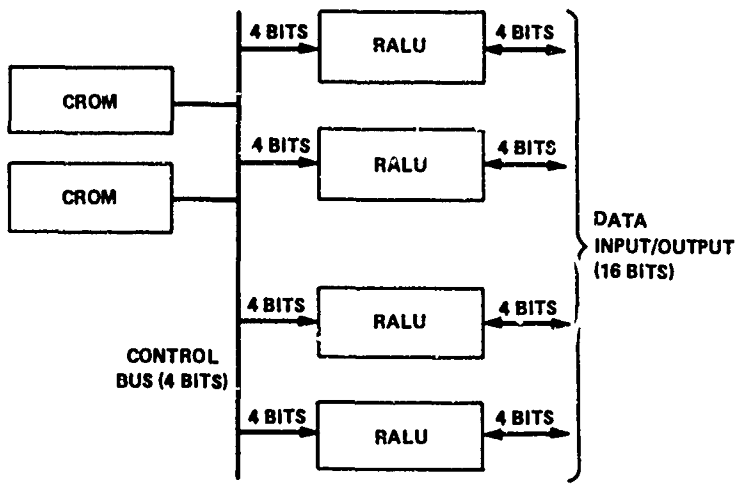

Figure 9. TDY-52B CPU Components

The Clock Generator in Figure 8 provides the CPU and external circuits to the TDY 52B with the required timing signals. There are eight time phases to each execution of an instruction resident in the CROM. Collectively the eight time phases are called one microcycle and a number of microcycles is required to execute each instruction resident in the external memory. Figure 10 shows the timing relationships. Time phase 4 (T4) may be extended during reading/writing operations when external memory requires slower access times than 525 ns. The use of external circuitry is required to extend T4 greater than two time phases. For more detailed information contact Teledyne Systems Company, Northridge, CA 91324.

Figure 11 is a flowchart of the TDY-52B operation starting with application of power. When power is first applied all RALU registers, flags and the LIFO stack are cleared to zero. The microprogram then enters an initialization sequence, in which the Program Counter (PC) is set to a starting value of $FFFE_{16}$, that is, the next-to-last location in the memory which is the first executed instruction.

2.4 Computer Housekeeping

The Computer Housekeeping circuit as shown in Figure 13 provides the TDY-52B with an approximate 10 ms interrupt and a system clear signal (SYSCLR). This circuit also provides the PCM and debounce clocks.

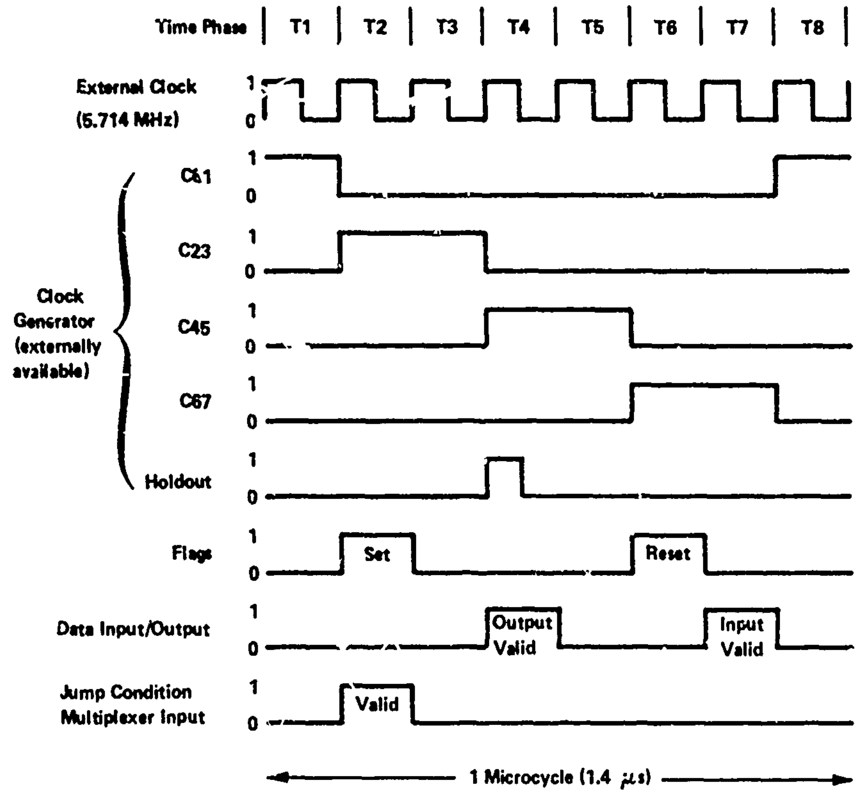

Figure 10. TDY-52B Timing Diagram

At system power turn on Reset is held momentarily low while Interrupt and Computer Start are held high. The TDY-52B will execute its first instruction from location $FFFE_{16}$ of the memory when Computer Start is momentarily grounded. To reset the TDY-52B while payload power is on momentarily ground Reset while momentarily turning OFF the TDY-52B minus 12 volt supply.

Interrupt Start is momentarily grounded during rocket liftoff enabling the interrupt clock. C45 in Figure 13 synchronizes the Interrupt Clock with the TDY-52B. Synchronization is required to prevent a positive transition during Time Phase 2 (T2, Figure 10). Interrupt Enable (INTEN) in Figure 13 is set by the TDY-52B under software control and cleared automatically upon the TDY-52B's recognition of a positive Interrupt Clock transition. Upon recognition of a positive Interrupt Clock transition the TDY-52B will halt and not execute the instruction from memory location 0001_{16} until the Interrupt Clock is zero, hence the use of the last D Flip-Flop in Figure 13.

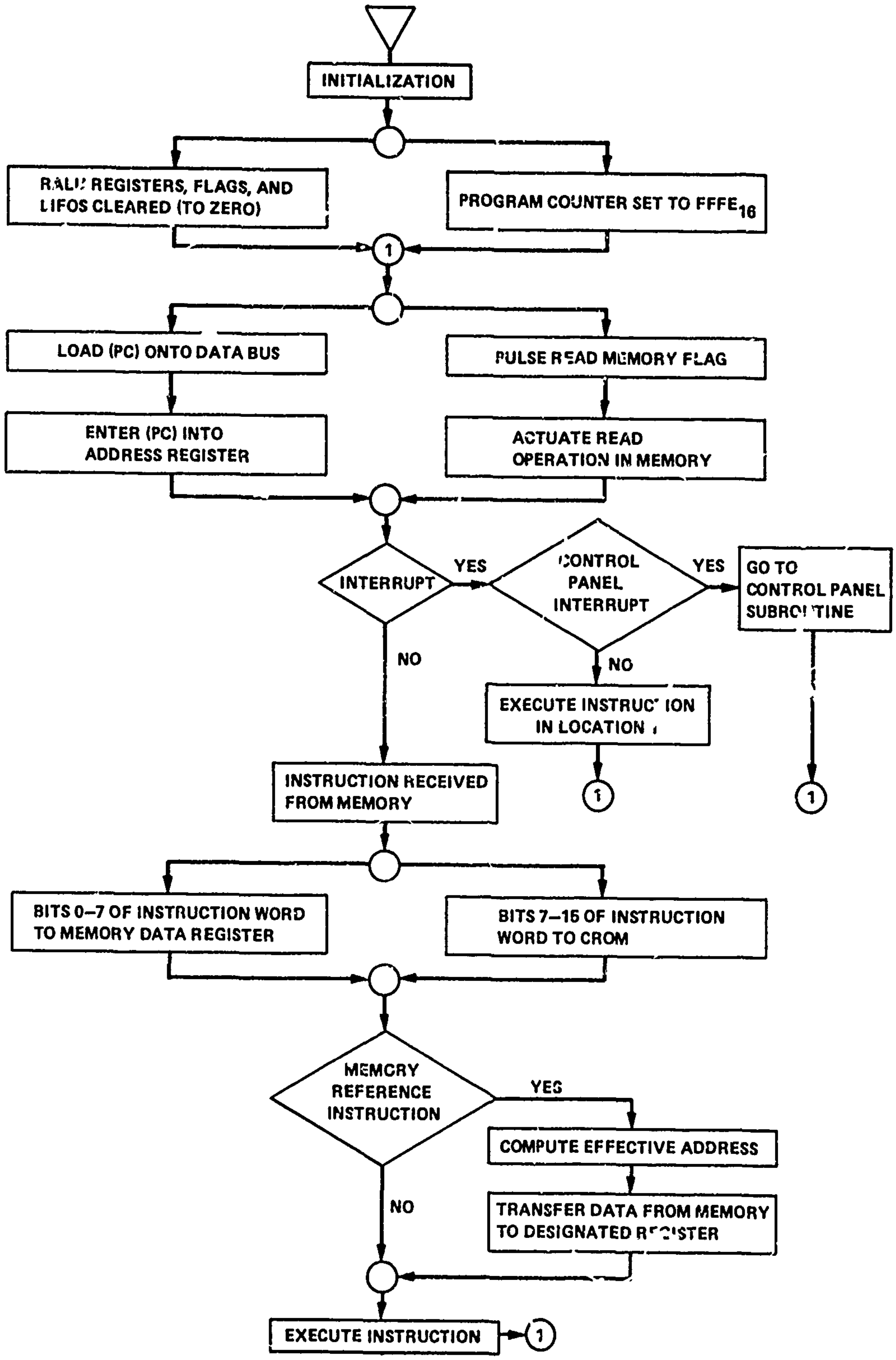

Figure 11. TDY-52B Operation Flowchart

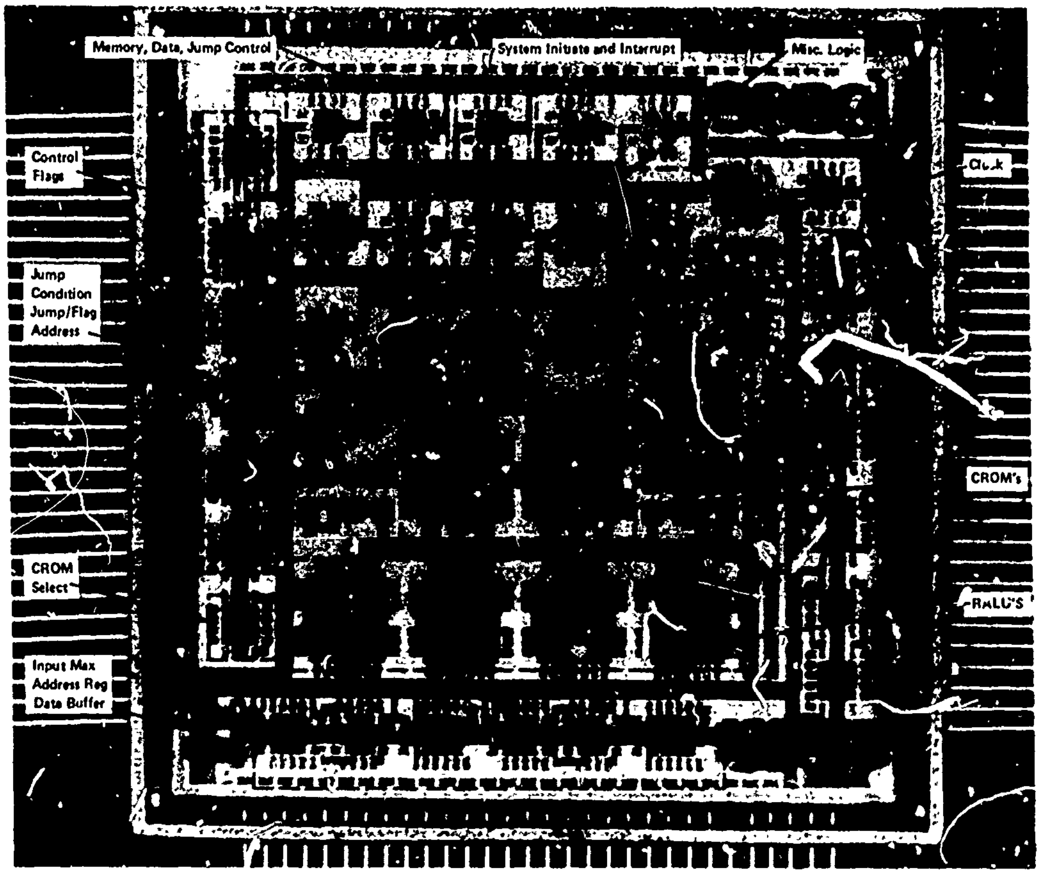

Figure 12. TDY-52B

2.5 Memory Interface

The Memory Interface circuit provides the necessary timing and control signals for interfacing of the memory with the TDY-52B. Figures 14 and 15 make up the Memory Interface Circuit while Figure 16 was used in place of Figure 14 during software development of the dispersion control system. See Figure 17 for circuit timing.

The Timing Interface Circuit of Figure 15 extends T4 to allow adequate access time to the CMOS RAM's. During a read microcycle, address information is sent out at T4 and the TDY-54B expects data back at T7 of the same microcycle. During a write microcycle data is sent to the memory during T4 of the next microcycle. The TDY-52B employs latches on its address lines eliminating the need to store address information externally. TDY-52B output signal Holdout triggers the Holdin signal high causing T4 to extend for 7.0 time phases. Holdout is generated only during a read/write (R/W) microcycle.

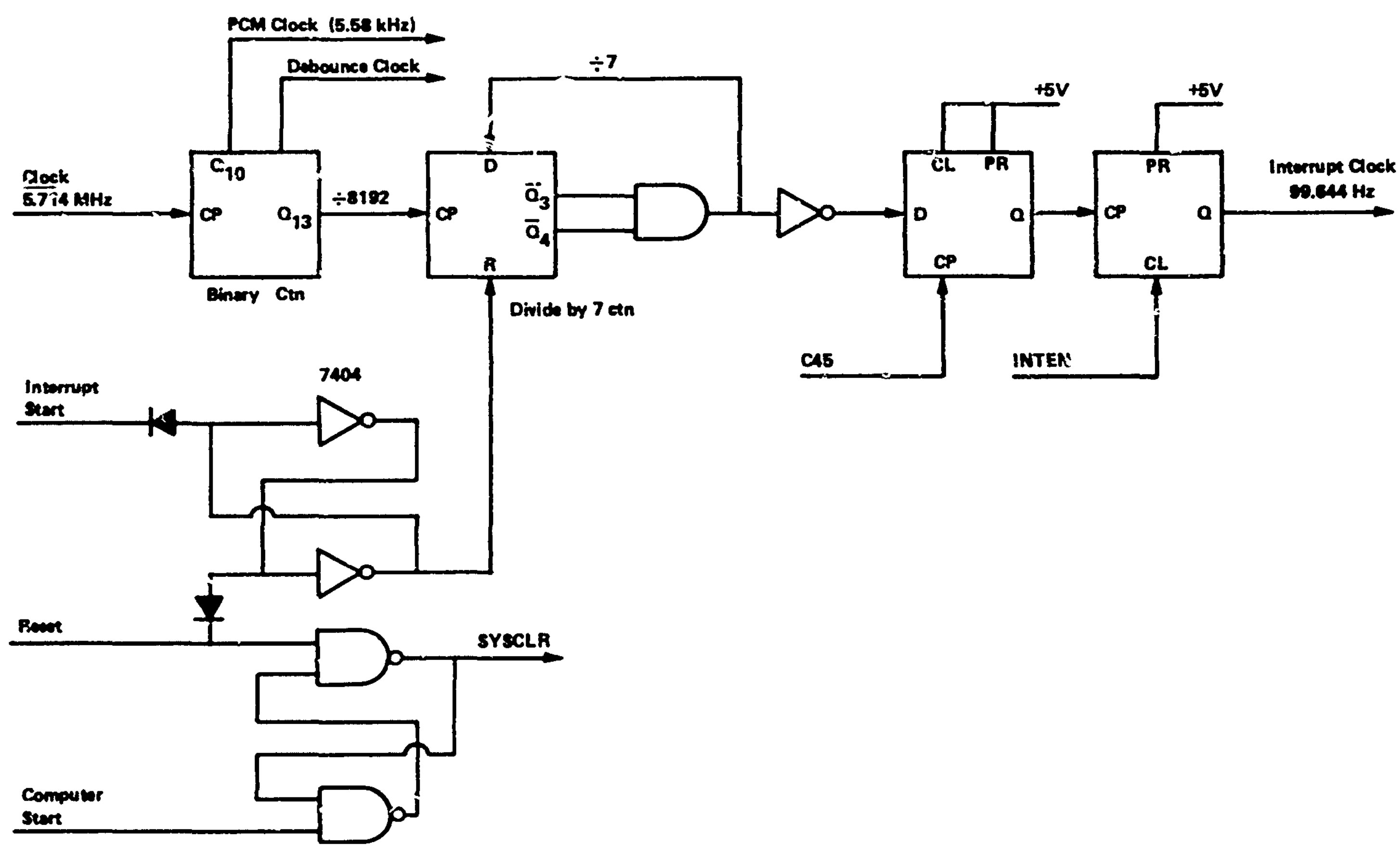

Figure 13. Computer Housekeeping

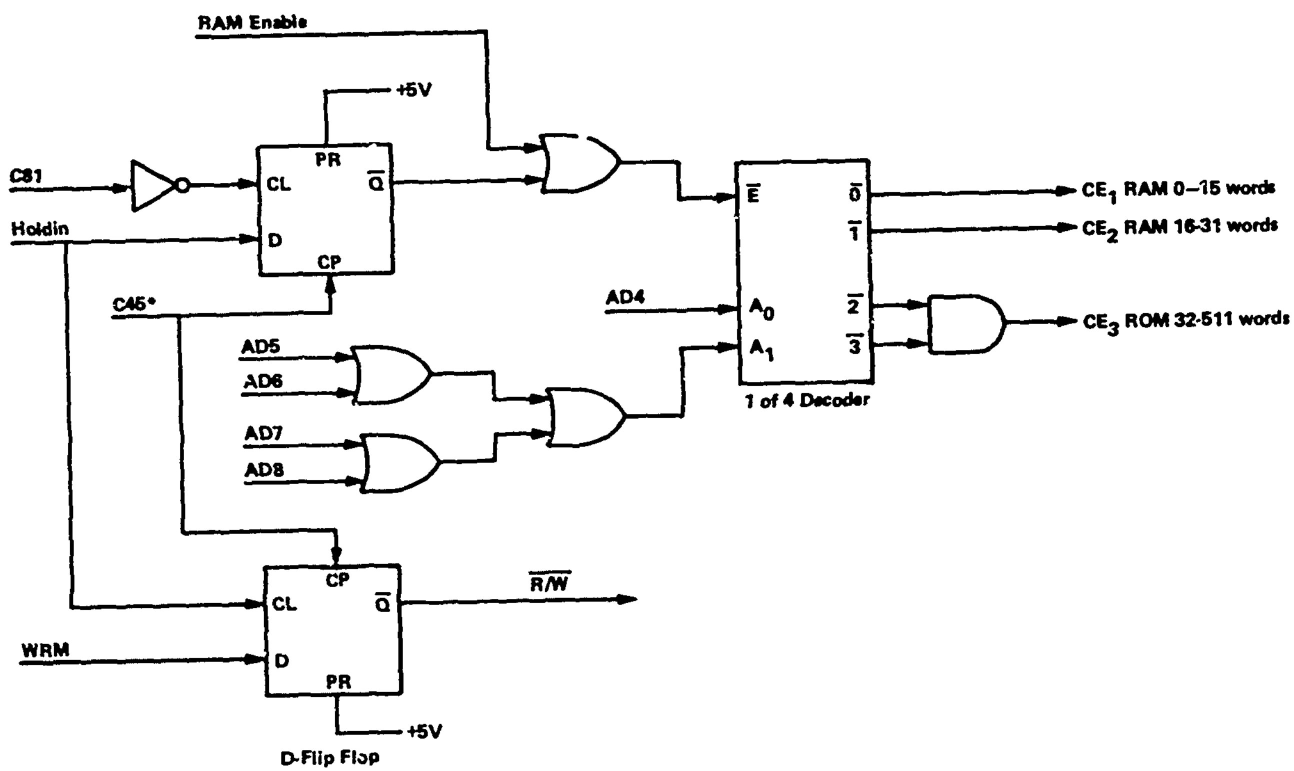

Figure 14. TDY-52B Interface Circuit to ROM/RAM Mix

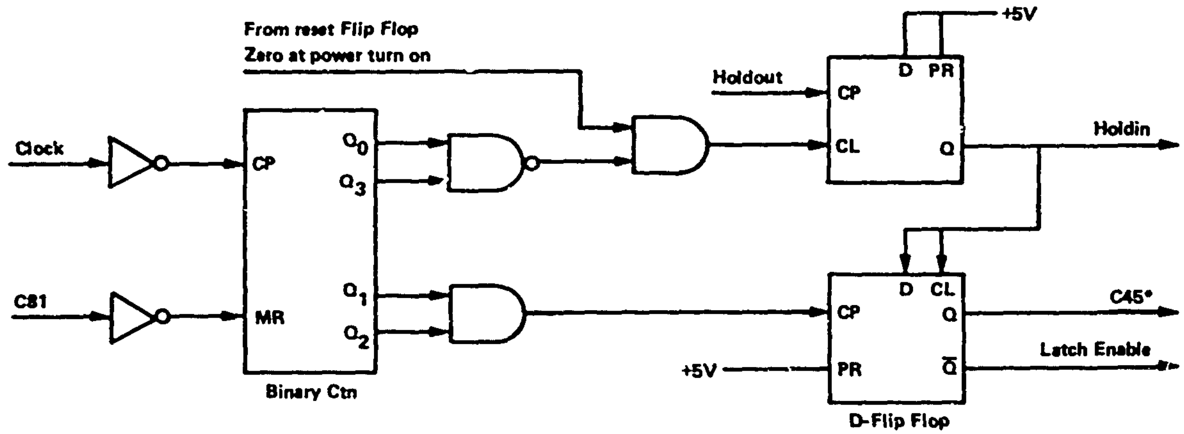

Figure 15. TDY-52B Timing Interface Circuit

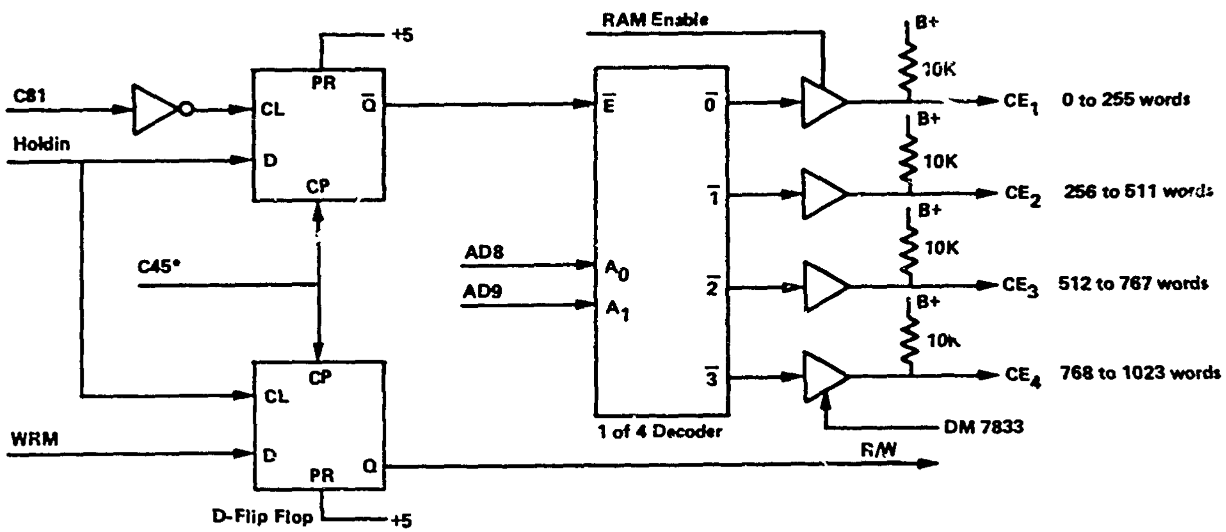

Figure 16. TDY-52B Interface Circuit to 4096 CMOS RAM

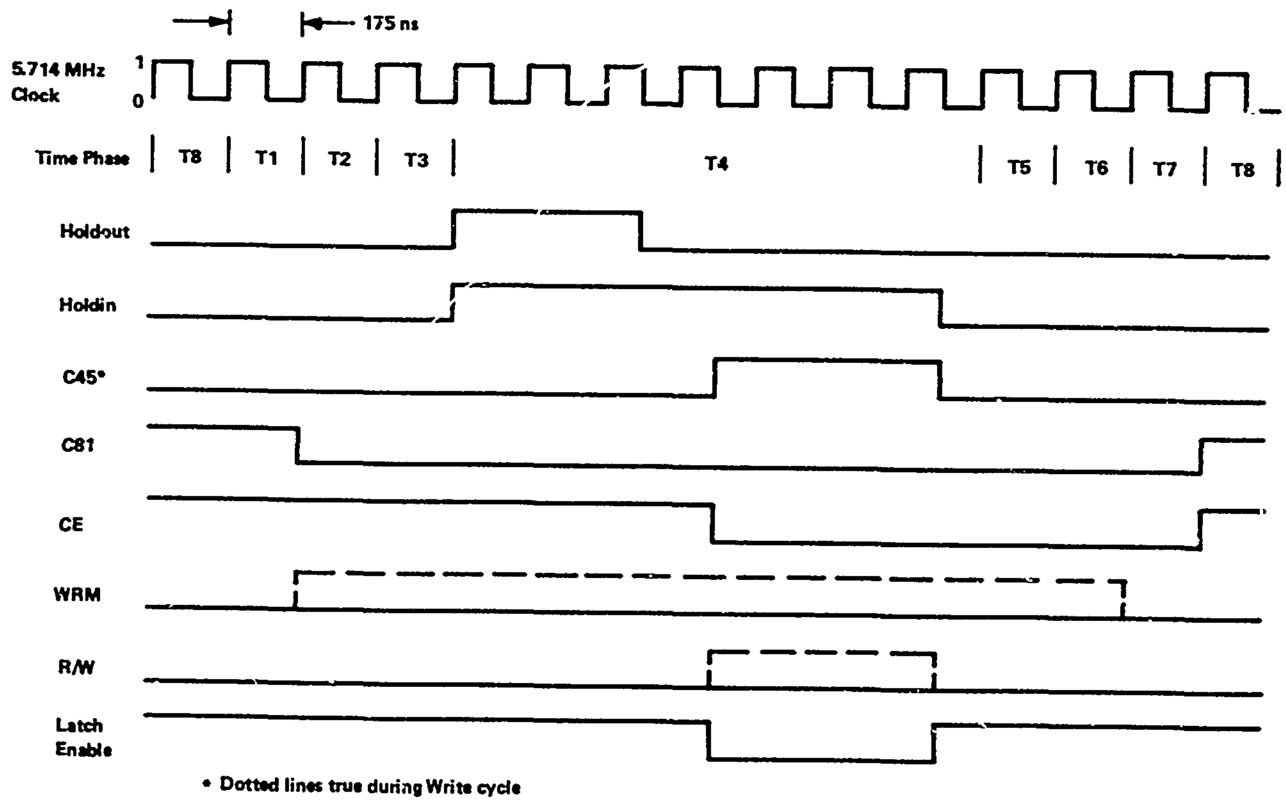

Figure 17. TDY-52B Timing for R/W Cycle

The circuits of Figures 14 and 15 provide their respective memories with Chip Enable (CE) and Read/Write (R/W) signals. Because of the large capacitive loading created by the CMOS RAM's, the TDY-52B address lines take approximately 500 ns to reach a stable state, the CE signal is delayed for 612.5 ns after the start of T4. The RAM Enable line in Figures 14 and 16 is used in conjunction with the ROM Loader, where a logic zero enables access to the memories.

3. DISPERSION CONTROL SOFTWARE

A detailed description of the Dispersion Control Software is contained in the following sections and a complete listing in Appendix C. Appendix B describes the TDY-52B Assembly Language used in Appendix C while Appendix E is a listing of the terms used by the flowcharts and listings.

3.1 Auto Pilot Difference Equation

Eq. (1) is the autopilot filter[1] chosen to control the Paiute-Tomahawk Rocket. The dispersion control system uses two of these auto pilots, that is, control in pitch plane and control in yaw plane. δc is a positional command in degrees to the cannards in either the pitch or yaw plane and θ_c is the error signal generated in the pitch or yaw planes.

Eq. (2) is written by matching[2] zeros and poles in the z-domain with those of Eq. (1) in the s-domain. The constant K_1 is determined by equating the final value of Eq. (1) with Eq. (2) in Eq. (3). Expanding Eq. (2) as shown in Eq. (4), the autopilot difference equation, Eq. (5) falls out.

Since fixed point arithmetic will be used by the software, Eq. (5) must be scaled. The maximum absolute allowable error signal is a 22.5° − .08789° step where $|\delta c| \leq 210.11$. The scale factor 256 will be used as shown in Eq. (6). The factor .8 is absorbed by K_2 to save a software multiplication step when converting the autopilot fixed point output for use by the D/A converters.

Eqs. (7) and (8) are the equations used by the software.

$$\frac{\delta c(s)}{\theta_c(s)} = \frac{K}{(s+w)^2} \qquad (1)$$

where K = 6 and w = .8,

1. Wilson, George, Martin Marietta Aerospace.
2. Technique from Software Research Corp.

$$\frac{\delta c(z)}{\theta_c(z)} = \frac{K_1}{(z - e^{-wT})^2}, \quad (2)$$

where T = sampling period.

$$\frac{K}{w^2} = \frac{K_1}{(1 - e^{-wT})^2} \text{ as } \begin{matrix} s \to 0 \\ z \to 1 \end{matrix}, \quad (3)$$

where $K_1 = K \left[\frac{1 - e^{-wT}}{w}\right]^2$.

$$\delta c = 2e^{-wT} Z^{-1} \delta c - e^{-2wT} Z^{-2} \delta c + K_1 Z^{-2} \theta_c, \quad (4)$$

$$\delta cn = 2e^{-wT} \delta cn\text{-}1 - e^{-2wT} \delta cn\text{-}2 + K_1 \theta cn\text{-}2, \quad (5)$$

$$\left[\frac{\delta cn}{256}\right] = 2 A \left[\frac{\delta cn\text{-}1}{256}\right] - B \left[\frac{\delta cn\text{-}2}{256}\right] + \frac{K_2}{256} [\theta cn\text{-}2], \quad (6)$$

where $K_2 = .8 * K_1 * 256 * .08789°$.

$A = e^{-wT}$
$B = e^{-2wT}$
$[\theta cn\text{-}2]$ = Integer quantity of .08789° steps which is presently divided by 256. Note that $[\theta cn\text{-}2] \neq \theta cn\text{-}2$.

$$ADP = 2 * A * ADP1 - B * ADP2 + \frac{K_2}{256} * THETA2, \quad (7)$$

$$ADY = 2 * A * ADY1 - B * ADY2 + \frac{K_2}{256} * PSIG2. \quad (8)$$

3.2 Auto Pilot Flow Chart

Figure 18 is a simplified flow chart of the Dispersion Control Software.

Execution begins with Block A (see Figure 19) where test commands are sent to the fins and the 32 word by 16 bit RAM is initialized. Upon completion of Block A the TDY-52B will wait for activation of the Interrupt Start Line. Every 10 ms after activation of the Interrupt Start, TN will be incremented and compared against TL (Launch Time), TI (Initiate Actuator Time), T13 (TI . 100 ms), TF (Lock Fins Time), TF3 (TF + 100 ms), and T2 (2nd stage Initiate Time), that is, variable TN will serve as the system clock. Block B will perform roll transformation of

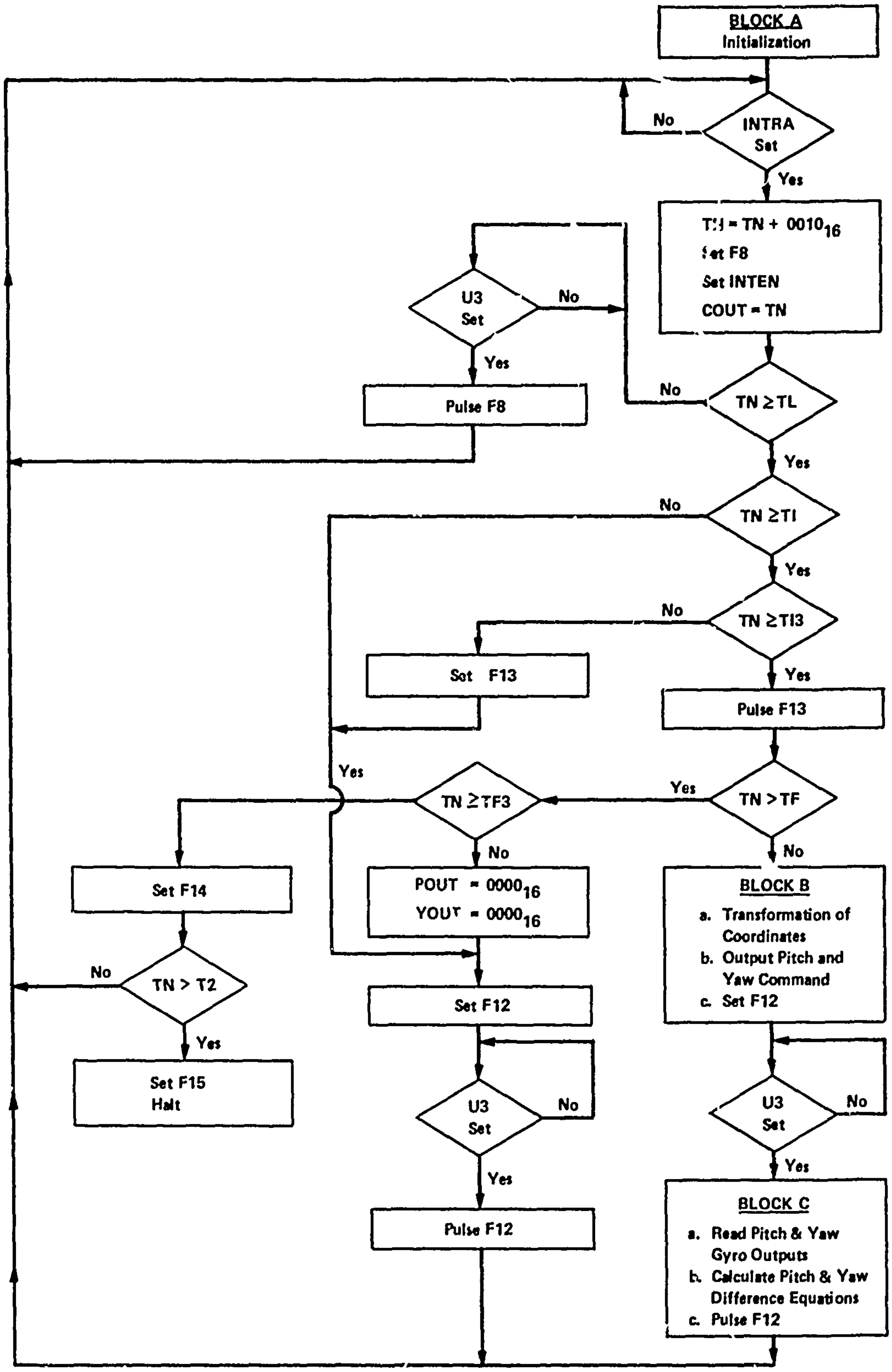

Figure 18. Auto Pilot

coordinates while Block C will calculate the difference equations. Note that Flags F8 and F12 and user jump input condition U3 were used to synchronize the TDY-52B with the AFCRL EAI 8900 during ground testing.

3.3 Block A

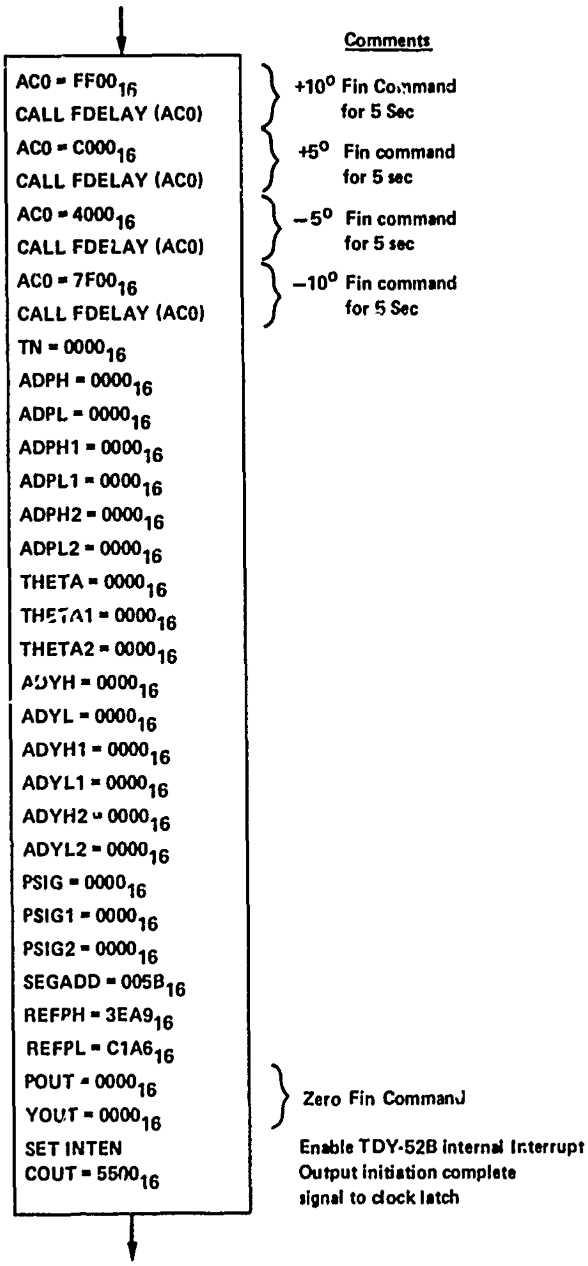

Figure 19. Block A

3.4 Block B

Block B calculates the effect of Gyro roll position phi, ϕ, on the pitch and yaw commands (see Figure 20).

Gyro roll position phi is loaded in AC0 using the format of Figure 21. AC0 is right justified by shifting AC0 5 times. AC2 is equated to AC0 and shifted twice to the right. The value in AC2 will be used as an address to select the correct $|\sin\phi|$ and $|\cos\phi|$ from Figure 22 while bits of 0 and 1 of AC0 determine the correct quadrant. The first If statement of Block B determines which of two pairs of quadrants, 1&3 or 2&4, is phi presently located. If phi is located in quadrants 1&3, AC2 is the address of $|\sin\phi|$ while 0020_{16} - AC2 is the address of $|\cos\phi|$. If phi is located in quadrants 2&4. 0040_{16} - AC2 is the address of $|\sin\phi|$ while AC2 0020_{16} is the address of $|\cos\phi|$. The last two If statements of Block B determine the signs of $|\sin\phi|$ and $|\cos\phi|$. Finally the roll transformation equations are calculated and pitch & yaw commands sent.

3.5 Block C

Block C calculates the pitch & yaw error signals and uses subroutine CALC (AC3) to calculate the pitch & yaw difference equations. Figure 23 is a flow chart of Block C.

Gyro pitch θ_g, and yaw, ψ_g, position is placed in AC0 using the format of Figure 24. Notice that this format has a scale factor of 512 instead of 256 which is used by the Auto pilot difference equations. Before subroutine CALC (AC3) is called the pitch & yaw error signals are multiplied by 2.

Pitch & yaw error signals are calculated by subtracting the gyro input position from the desired reference position. Gravity turn affects the pitch reference, Figure 25, while the yaw reference is a constant 4000_{16}. The pitch reference is calculated by successive summations rather than time consuming multiplications as shown in Eqs. (9) and (10).

$$\text{Pitch Ref} = at + b \tag{9}$$

where a = slope
t = time
b = intercept.

$$\text{Pitch Ref} = \text{REFPN} + \Sigma\ \text{ATNH} \tag{10}$$

where REFPN = $b + a_n t_n$
t_n = time line segment n, Figure 25, begins n = 1, 2, or 3
ANTH = $a_n * T$
a_n = slope of line segment n
T ≈ 10 ms sampling rate

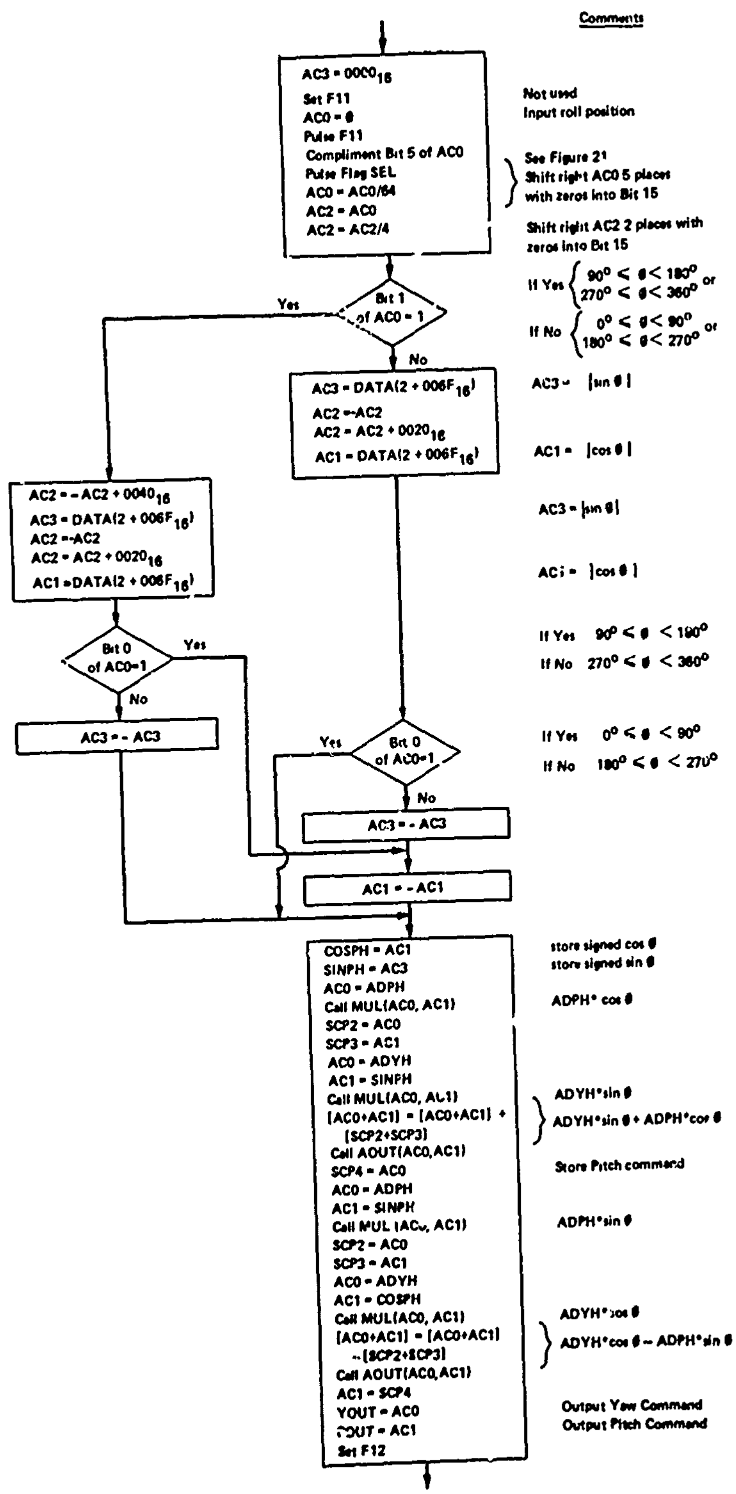

Figure 20. Block B

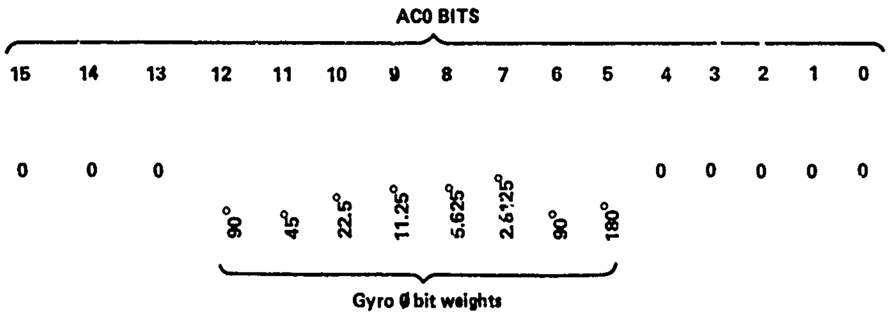

Figure 21. Input Format of Gyro ϕ

AC2	Quadrant 1 & 3		Quadrant 2 & 4		Magnitude
	|Sin ϕ|	|Cos ϕ|	|Sin ϕ|	|Cos ϕ|	
0 0 0 0	Sin 0°, Sin 180°	Cos 90°, Cos 270°	Sin 180°, Sin 360°	Cos 90°, Cos 270°	0 0 0 0
0 0 0 1	↓	↑	↑	↓	0 6 4 7
0 0 0 2					0 C 8 B
0 0 0 3					1 2 C 8
0 0 0 4					1 8 F 8
0 0 0 5					1 F 1 9
0 0 0 6					2 5 2 8
0 0 0 7					2 B 1 F
0 0 0 8					3 0 F B
0 0 0 9					3 6 B A
0 0 0 A					3 C 5 6
0 0 0 B					4 1 C E
0 0 0 C					4 7 1 C
0 0 0 D					4 C 3 F
0 0 0 E					5 1 3 3
0 0 0 F					5 5 F 5
0 0 1 0	Sin 45°, Sin 225°	Cos 45°, Cos 225°	Sin 135°, Sin 315°	Cos 135° Cos 315°	5 A 8 2
0 0 1 1	↓	↑	↑	↓	5 E D 7
0 0 1 2					6 2 F 2
0 0 1 3					6 6 C F
0 0 1 4					6 A 6 D
0 0 1 5					6 D C A
0 0 1 6					7 0 E 2
0 0 1 7					7 3 B 5
0 0 1 8					7 6 4 1
0 0 1 9					7 8 8 4
0 0 1 A					7 A 7 D
0 0 1 B					7 C 2 9
0 0 1 C					7 D 8 A
0 0 1 D					7 E 9 D
0 0 1 E					7 F 6 2
0 0 1 F					7 F D 8
0 0 2 0	Sin 90°, Sin 270°	Cos 0°, Cos 180°	Sin 90°, Sin 270°	Cos 180°, Cos 360°	7 F F F

Figure 22. Absolute Value of Sin ϕ and Cos ϕ vs AC2

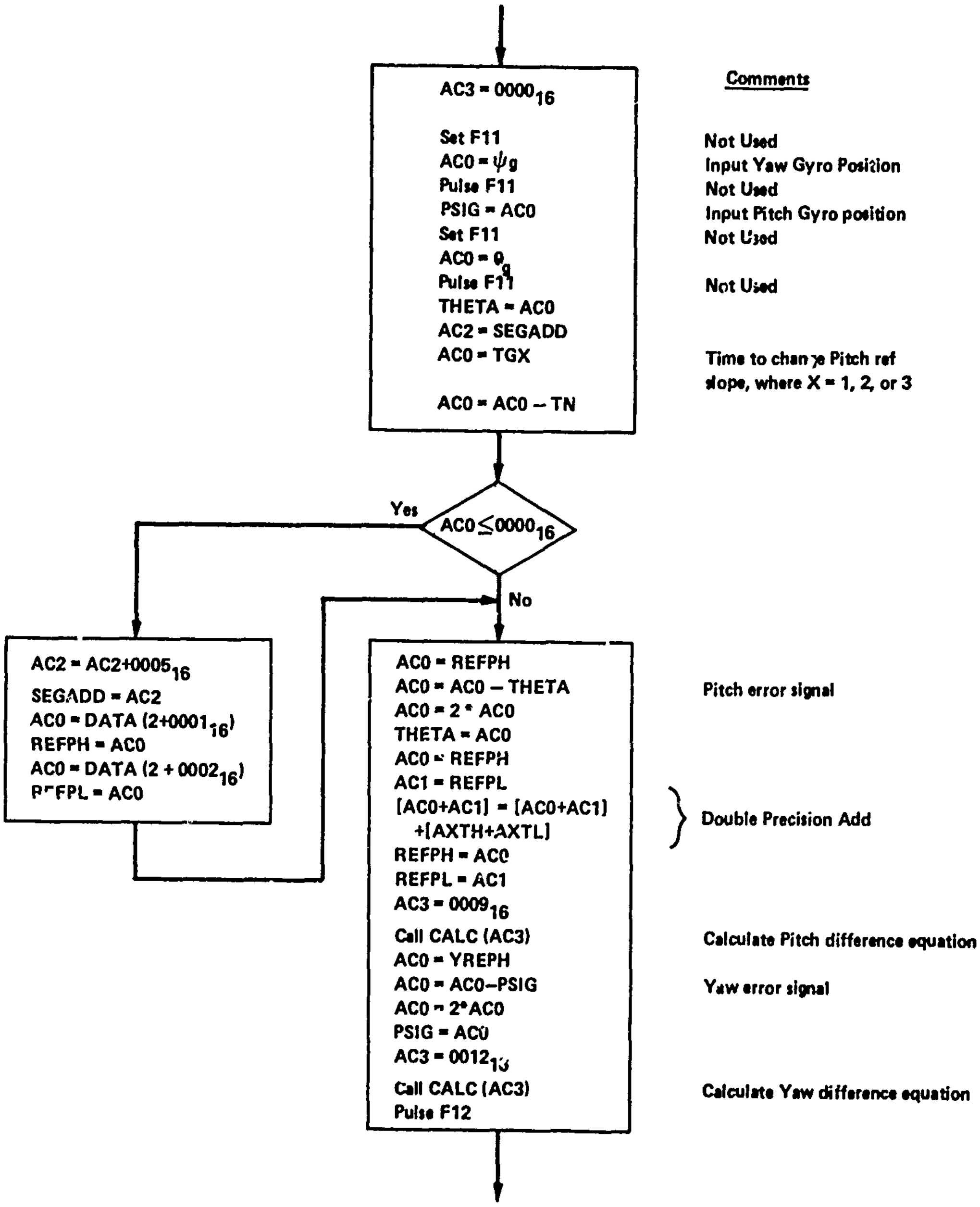

Figure 23. Block C

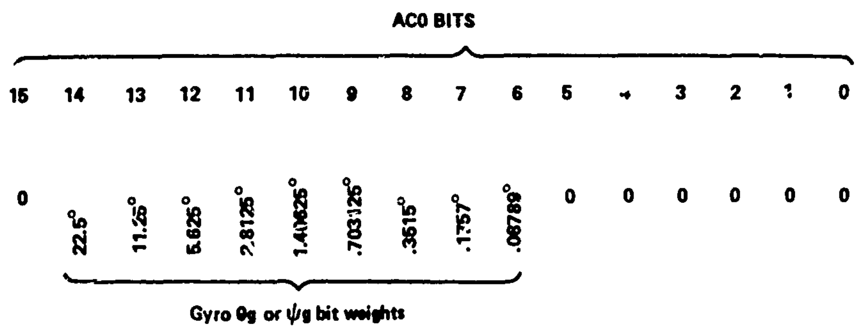

Figure 24. Input Format of Gyro θg and ψg

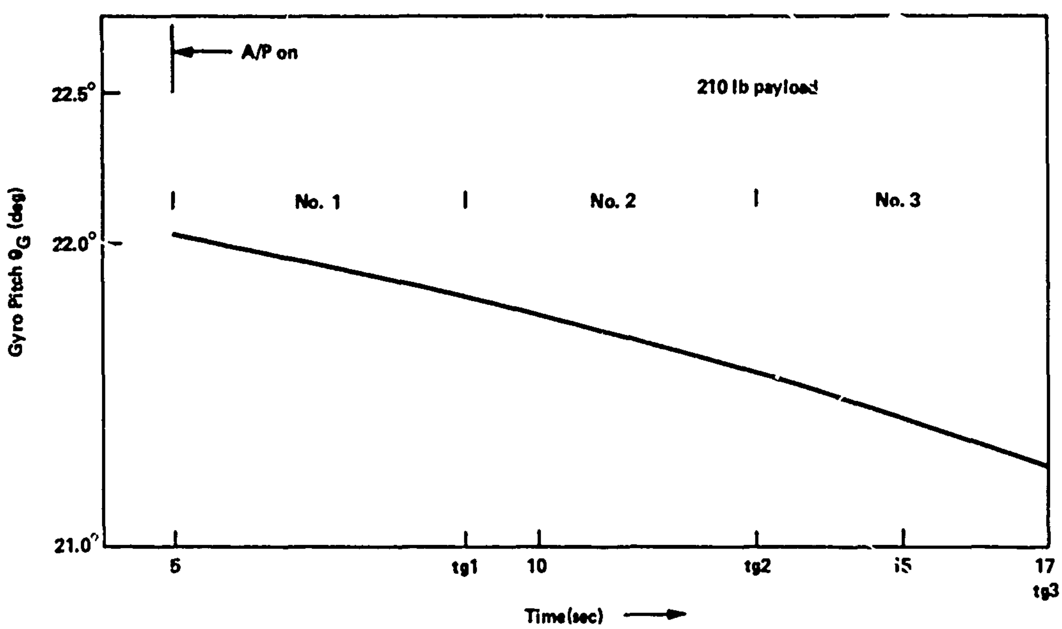

Figure 25. Pitch Gravity Turn – Paiute Tomahawk

3.6 Subroutines

3.6.1 SUBROUTINE MUL (AC0, AC1)

Subroutine MUL (AC0, AC1) calculates one-half the signed product of AC0 and AC1. The 32 bit result is placed in AC0 and AC1 with the Most Significant Part, MSP, in AC0 (see Figure 26).

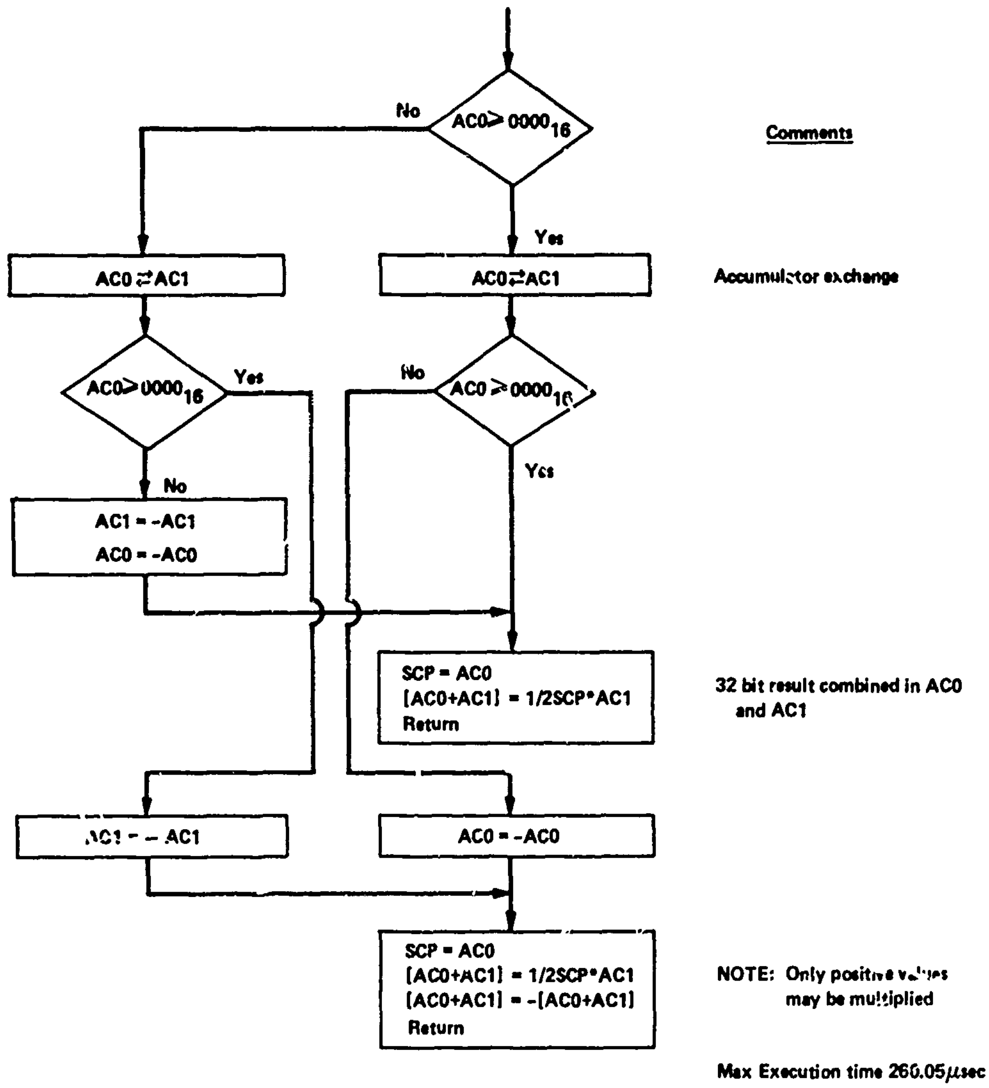

Figure 26. Subroutine MUL (AC0, AC1)

3.6.2 SUBROUTINE AOUT (AC0, AC1)

Subroutine AOUT (AC0, AC1) will convert the fin command in AC0 to match the format of the D/A converter (see Figure 27).

The first IF statement tests the sign bit of AC0 and sets AC3 accordingly. A test is then made of the fin command magnitude to determine if it exceeds the 10° maximum. If greater than 10°, AC0 is set to 10° and if less AC0 is multiplied by

128 removing in effect the scale factor used during its calculation. Remember that the factor .8 was buried in K2 of the difference equation, Eq. (6), so that only a 7 bit left shift is required at this point saving valuable time. The final IF statement determines the correct sign bit for AC0.

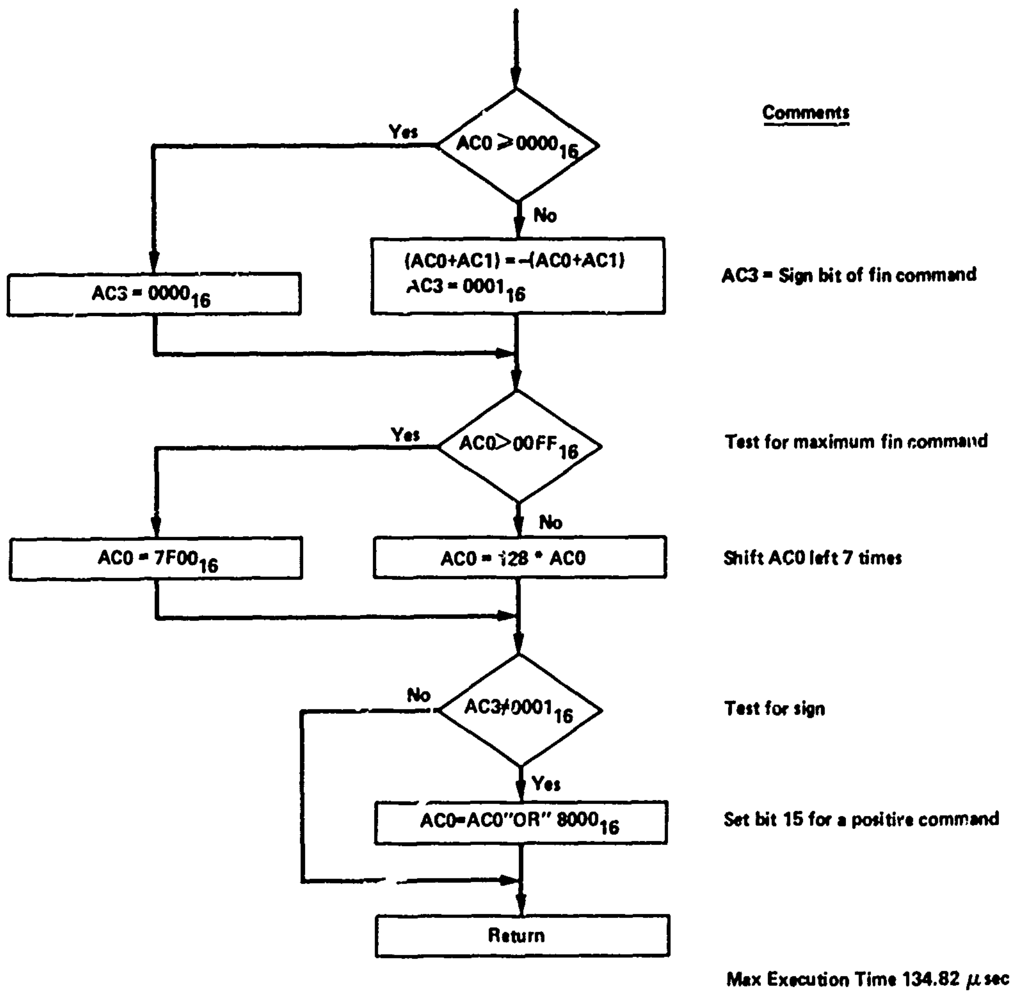

Figure 27. Subroutine AOUT (AC0, AC1)

3.6.3 Subroutine CALC (AC3)

Subroutine CALC (AC3) calculates the Auto Pilot Difference Equation, (see Figure 28). AC3 is used by the main program to tell the subroutine if the pitch or yaw difference equation is to be calculated.

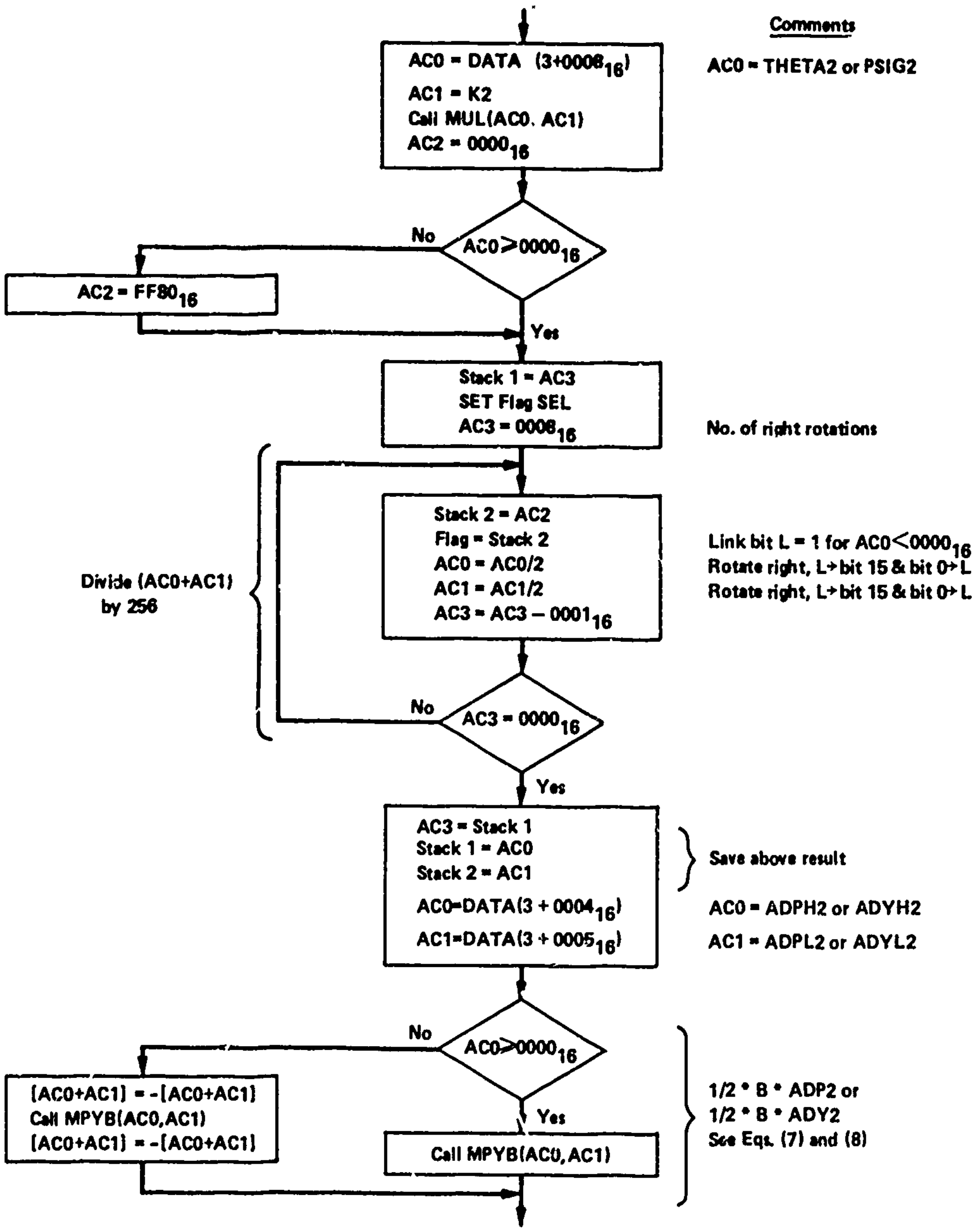

Figure 28. Subroutine CALC (AC3)

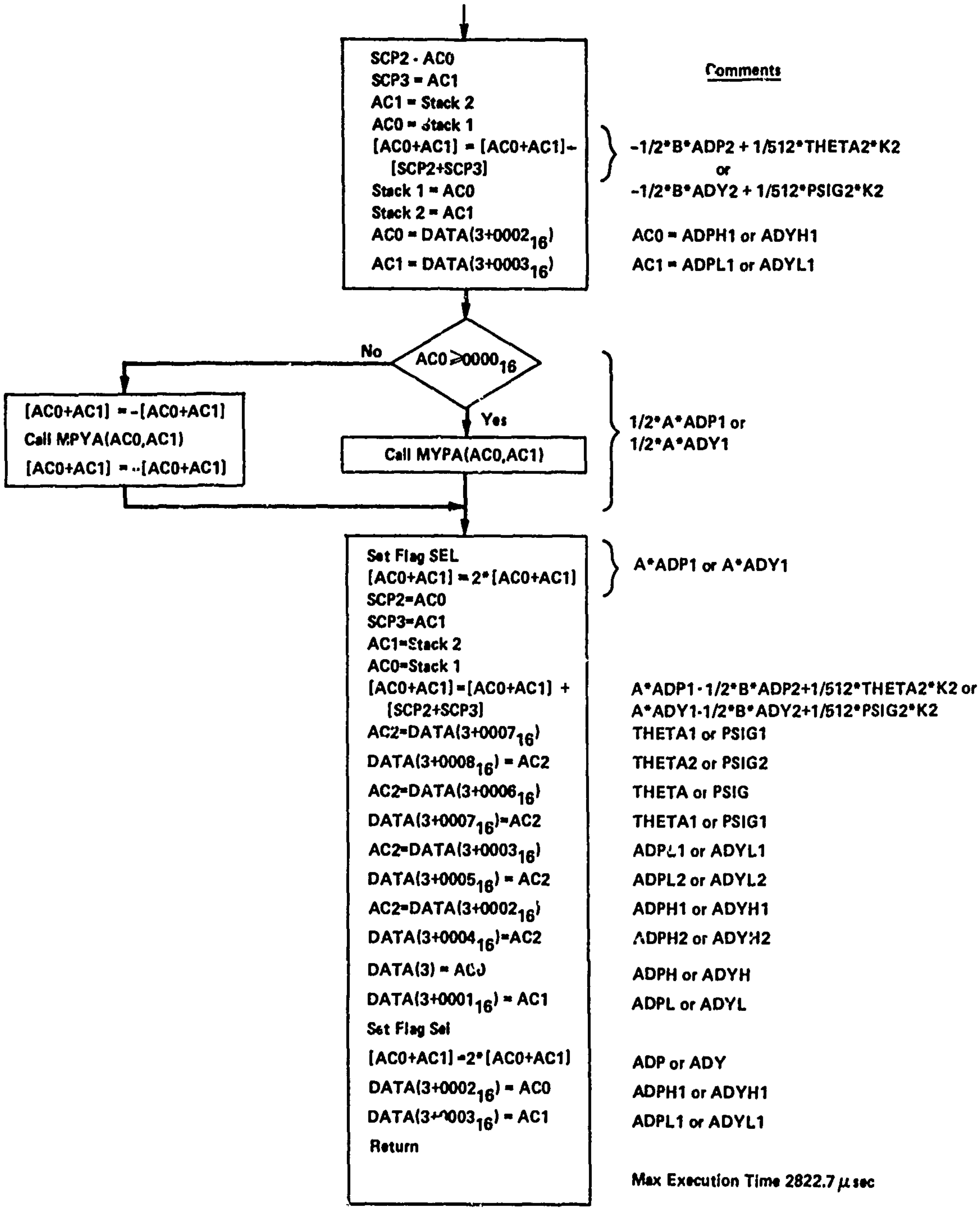

Figure 28. Subroutine CALC (AC3) (Cont)

3.6.4 SUBROUTINE MPYA (AC0, AC1)

Subroutine MPYA (AC0, AC1) calculates one-half the product of the 32 bit fixed point number formed by AC0 and AC1 and the 32 bit fixed point number formed by A and AL. The 32 bit result is placed in AC0 and AC1. MSP is in AC0 (see Figure 29).

3.6.5 SUBROUTINE MPYB (AC0, AC1)

Same as MPYA except the 32 bit fixed point number formed by B and BL is used.

Step	Comments
Stack1 = AC0	Store Most Significant Part (MSP)
Stack2 = AC1	Store Least Significant Part (LSP)
AC1 = A	
SCP = AC0	
[AC0+AC1] = 1/2*AC1*SCP	
SCP2=AC0 SCP3=AC1	Store 1/2 MSP*A
AC0=Stack 2	AC0 = LSP
Pulse Flag SEL AC0= 1/2 AC0	Shift AC0 right, bit 15 = 0
AC1 = A	
SCP= AC0	
[AC0+AC1] = 1/2*AC1*SCP	1/4 LSP*A
Set Flag SEL [AC0+AC1] = 2*[AC0+AC1]	1/2 LSP*A
AC1 = AC0	
AC0 = 0000_{16}	
[AC0+AC1] = [AC0+AC1] + [SCP2 + SCP3]	
SCP2 = AC0 SCP3 = AC1	Store 1/2MSP*A+1/2 LSP*A
AC0 = AL	
Pulse Flag SEL AC0 = 1/2 AC0	AC0 = 1/2 AL
AC1 = Stack 1	AC1 = MSP
SCP = AC0	
[AC0+AC1] = 1/2*AC1*SCP	1/4 MSP * AL
Set Flag SEL [AC0+AC1] = 2*[AC0+AC1]	1/2 MSP * AL
AC1=AC0	
AC0=0000_{16}	
[AC0+AC1] = [AC0+AC1] + [SCP2+SCP3]	1/2MSP*A+1/2LSP*A+1/2MSP*AL
Return	

Max Execution Time 815.5 μsec

Figure 29. Subroutine MPYA (AC0, AC1)

3.6.6 SUBROUTINE FDELAY (AC0)

Subroutine FDELAY (AC0) will provide an approximate 5 sec command to both the pitch and yaw fins. AC0 is the signed magnitude of the command and must be in the correct format for the D/A converters (see Figure 30).

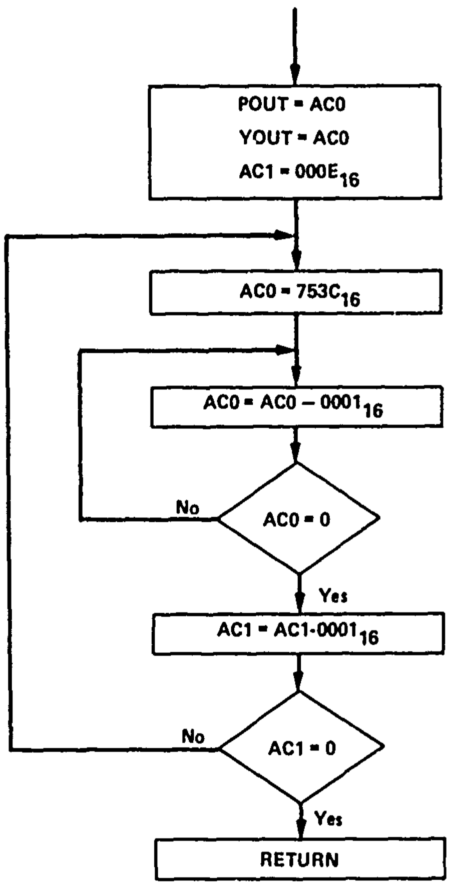

Figure 30. Subroutine FDELAY (AC0)

4. ROM LOADER

The ROM Loader is designed to edit or debug a program stored in the CMOS RAM. An ASR-33 and Figures 33 and 34 comprise the ROM Loaders hardware. There are three functions which the ROM Loader performs; List memory contents, Read a paper tape into the memory and Write into the memory from the teletype keyboard.

4.1 Operation

To use the ROM Loader switch S1 of Figure 34 must be in position one before starting the TDY-52B. When the TDY-52B is started the teletype will respond with the message "COMMAND:" and wait for one of three commands typed on the keyboard L, R or W:

L, AAAA, BBBB* typed on the teletype will produce a listing of the memory contents at the four digit hexadecimal location AAAA through location BBBB. See Figure 31 for sample output.

R typed on the teletype will start the paper tape reader. Figure 32 is the required papertape format.

```
COMMAND: L, 0020, 0024
 LOC     DATA

 0020    4700
 0021    4C10
 0022    C002
 0023    A002
 0024    0800
```

Figure 31. Sample ROM Loader Listing

```
               SPROCKET
00110    000  } Starting address of data in
00110    000  } ASCII coded hexadecimal
00110    010  } for this example 0020 (base 16)
00110    000  }
DDDDD    DDD  } DATA in ASCII coded
DDDDD    DDD  } hexadecimal, that is, for each
DDDDD    DDD  } memory location there will be
DDDDD    DDD  } four lines of punched tape.
  .           } MSD is first. The ROM Loader
  .           } increments the starting address
  .           } for the next block of data.
  .           } Note that all characters not in
  .           } Figure 38 will be ignored by
DDDDD    DDD  } Subroutine RECV (AC0, AC1)
00100    001    Exclamation mark terminates reading
                of the paper tape.
```

Figure 32. ROM Loader Paper Tape Format

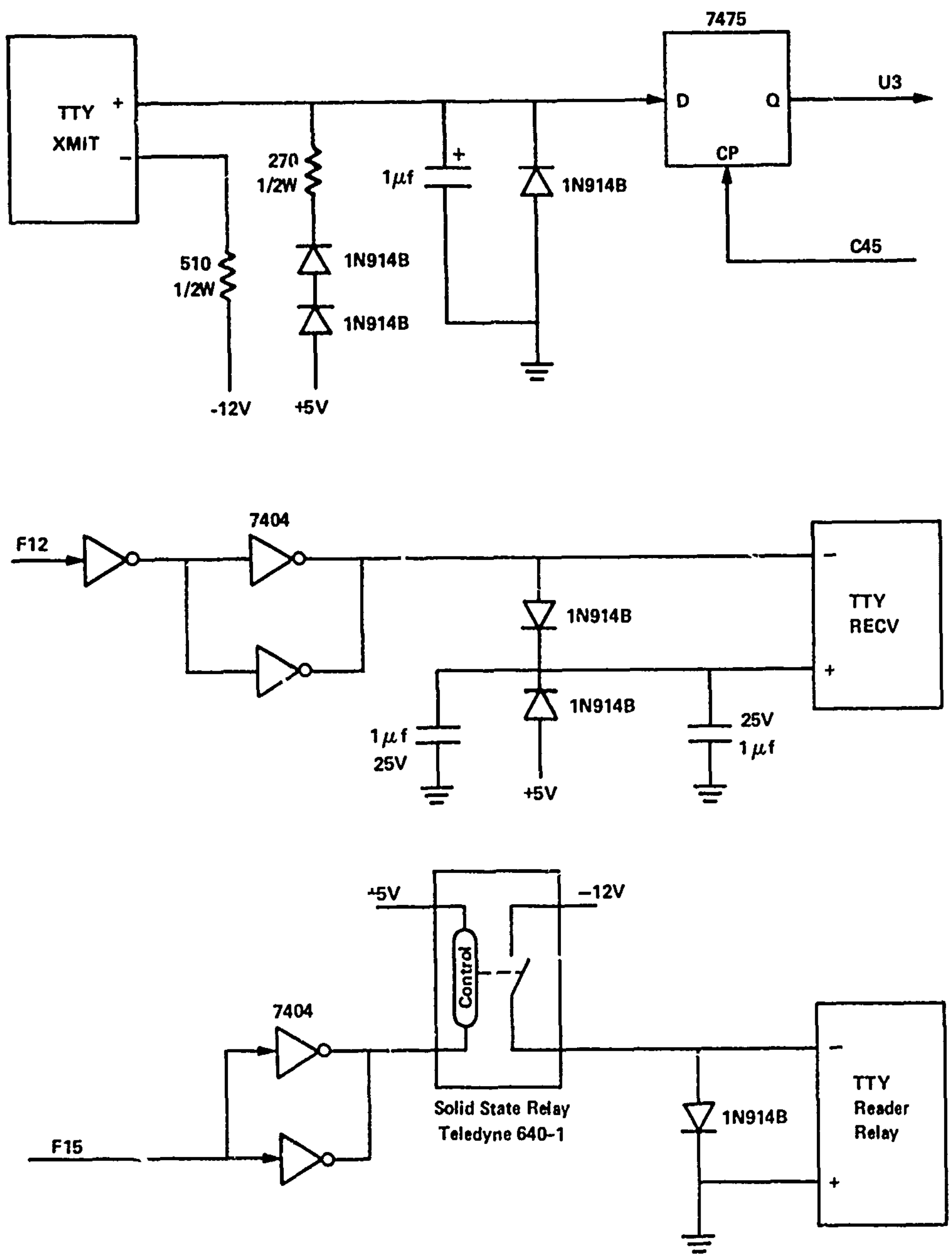

Figure 33. TTY Interface

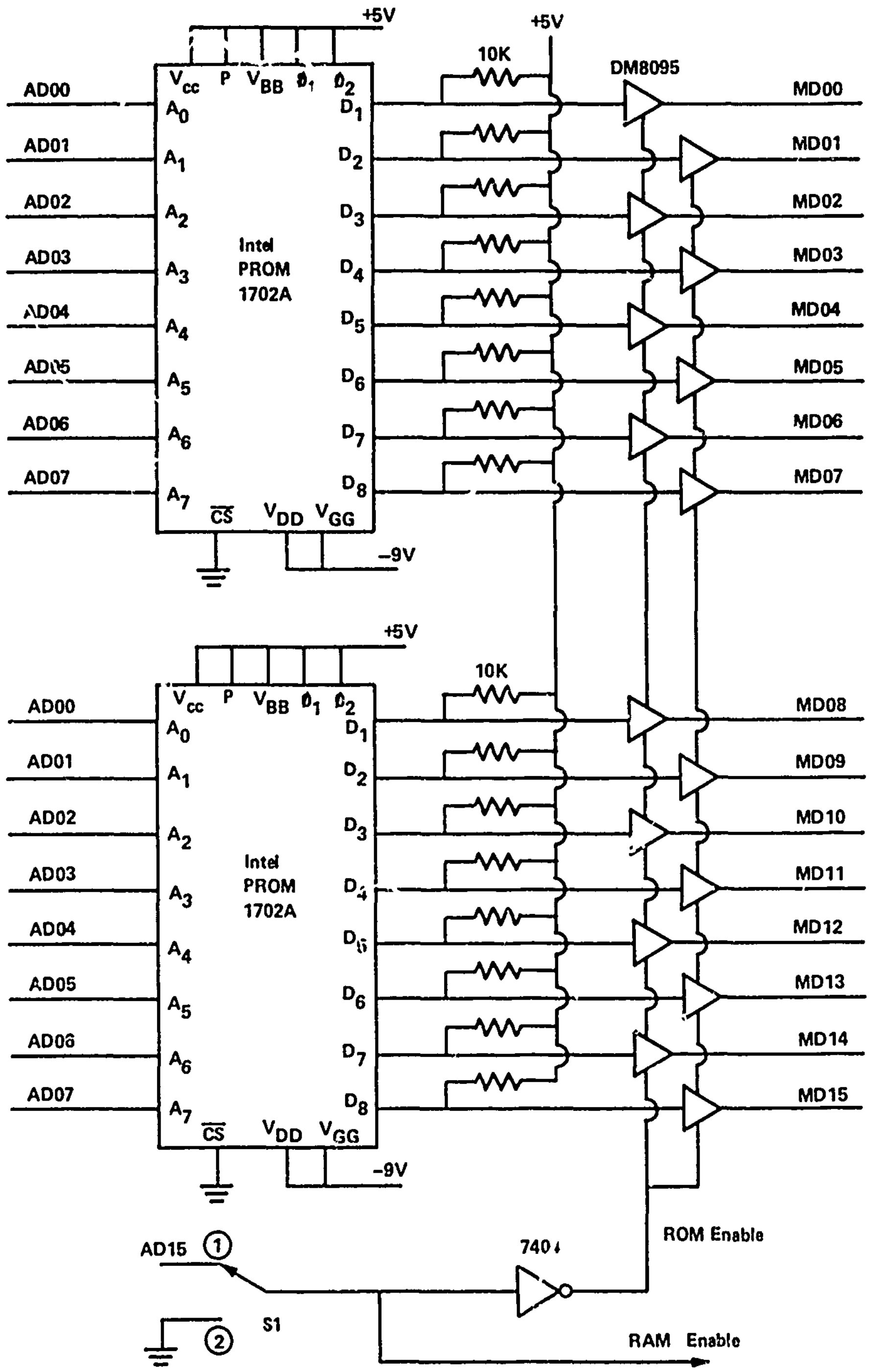

Figure 34. ROM Loader Interface

W, AAAA, BBBB* typed on the teletype will write the four digit hexadecimal value BBBB in memory location AAAA.

After completing each command the message "COMMAND:" is once again sent by the ROM Loader indicating it is ready for a new command. *Note commas in the above commands are not necessary. Figure 35 is the ROM Loader Flow Chart.

4.2 ROM Loader Electronics

The circuit diagrams in Figures 33 and 34 comprise the ROM Loader Electro[illegible]s. The two 256 word by 8 bit PROM's contain the ROM Loader software. Switch S1 of Figure 34 determines which memory the TDY-52B will fetch its first instruction from, that is, position 1 will enable the ROM Loader while position 2 will enable the Dispersion Control memory. The RAM Enable of Figure 34 is connected as shown in Figure 14, or when using the 4096 RAM in Figure 16. Address lines AD00 through AD07 and AD15 of the TDY-52B are used to address the ROM Loader memory (see Figure 34). Isolation between the ROM Loader Memory and the Dispersion Control Memory is accomplished by means of Tri-State outputs, which must be provided externally for the Intel PROM's, that is, the Hex Tri-State Buffers DM8095. MD00 through MD15 are the memory data input lines of the TDY-52B.

4.3 Subroutines

4.3.1 SUBROUTINE RECV (AC0, AC1)

Subroutine RECV (AC0, AC1) provides the necessary decoding of the 8 bit data words sent serially by the teletypes transmitter, (see Figure 36). The subroutine will return from where it was called with AC0 containing the decoded teletype data and AC1 containing the position of the teletype data in the look up table, Figure 38.

Upon recognition of a teletype start bit a 13.5 ms delay is intiated to wait for the middle of the first teletype data bit, (see Figure 37). After AC0 has been filled with the first seven data bits and the parity bit, bit 8, ignored, AC0 is compared against the look up table in Figure 38 to determine what alphanumeric character the teletype sent. If there is no match the subroutine will return to its beginning to wait for another teletype transmission. If AC0 contains the alphanumeric characters 0-F it is decoded from seven bits to four bits, that is, the four LSB's of AC0 will contain the hexadecimal digits 0-F with all other bits zero.

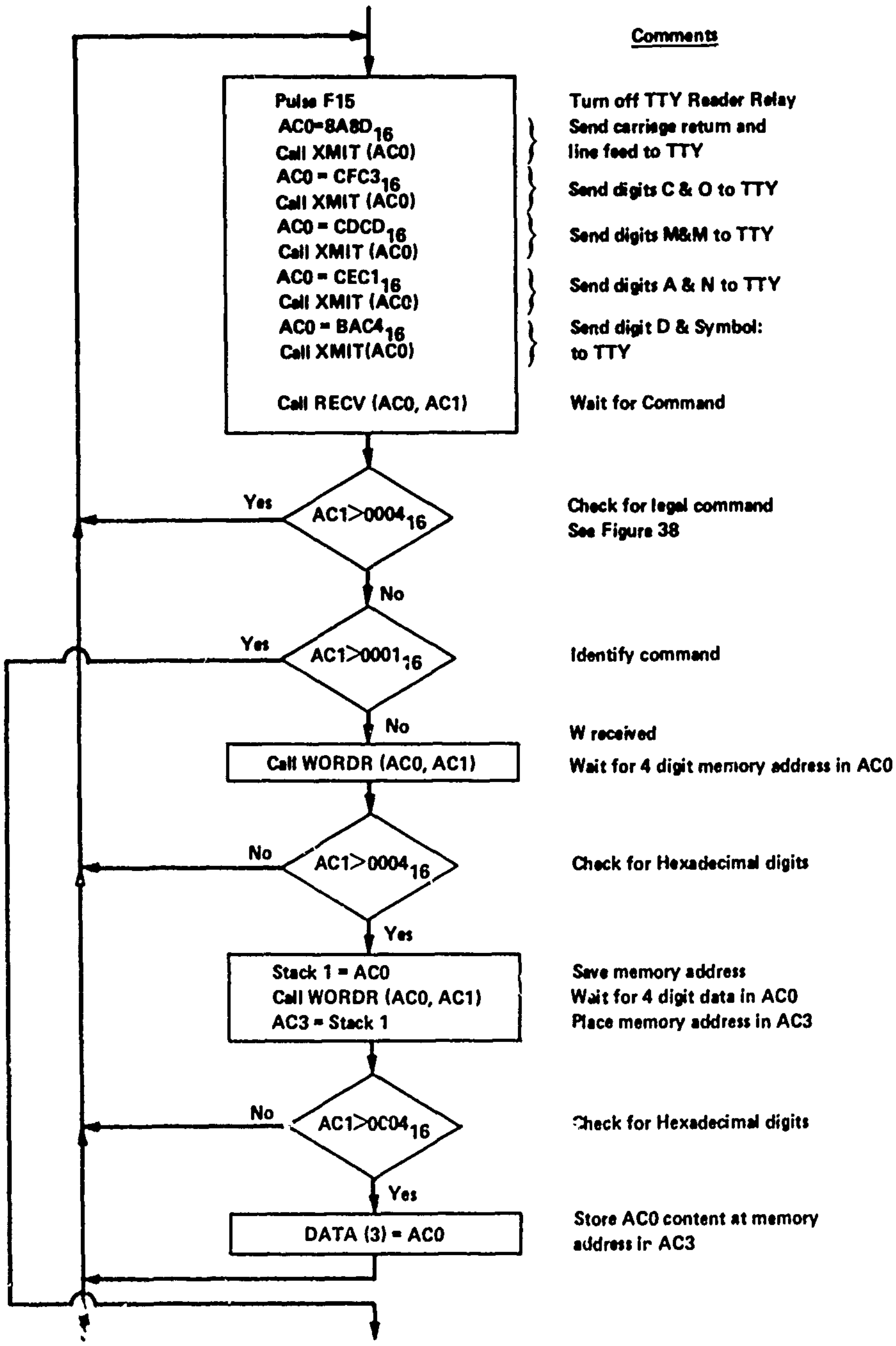

Figure 35. ROM Loader Flow Chart

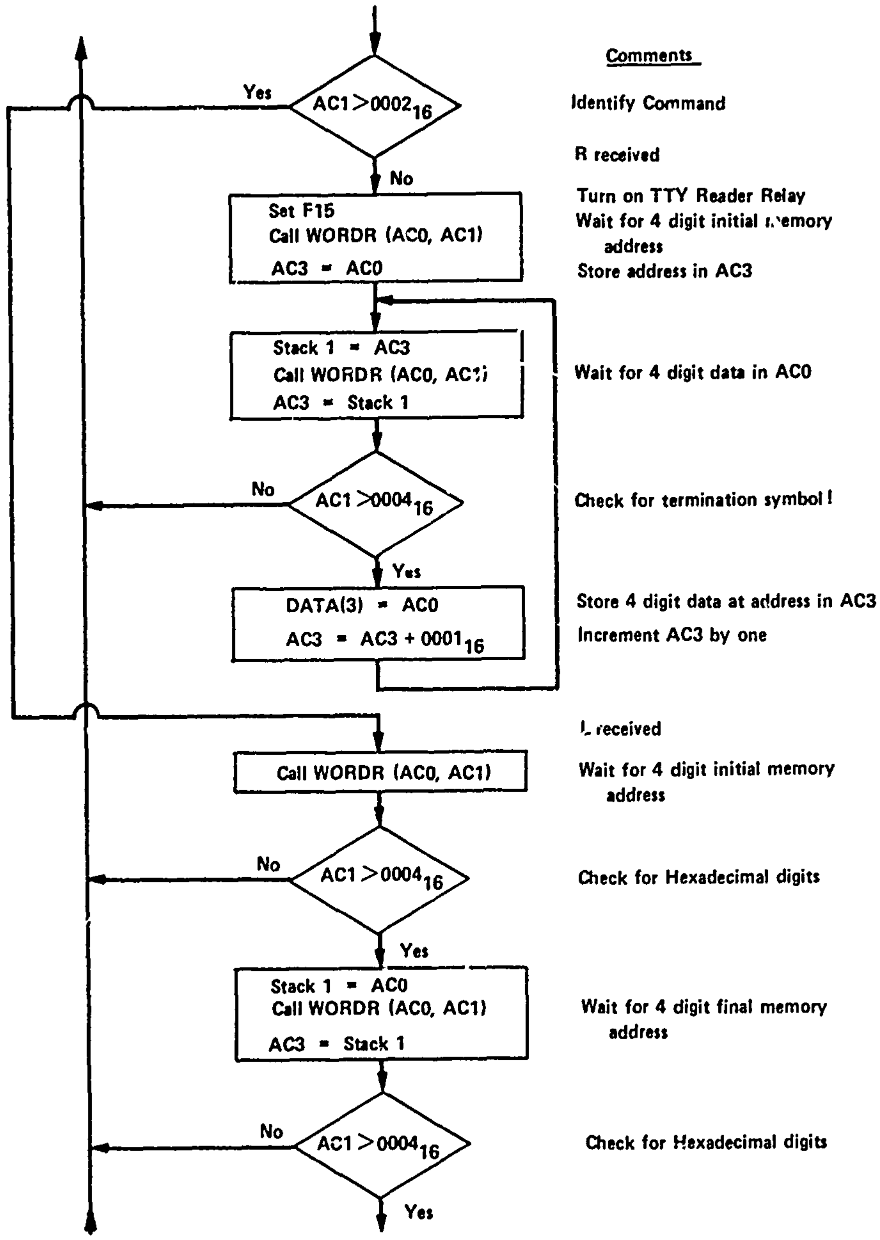

Figure 35. ROM Loader Flow Chart (Cont)

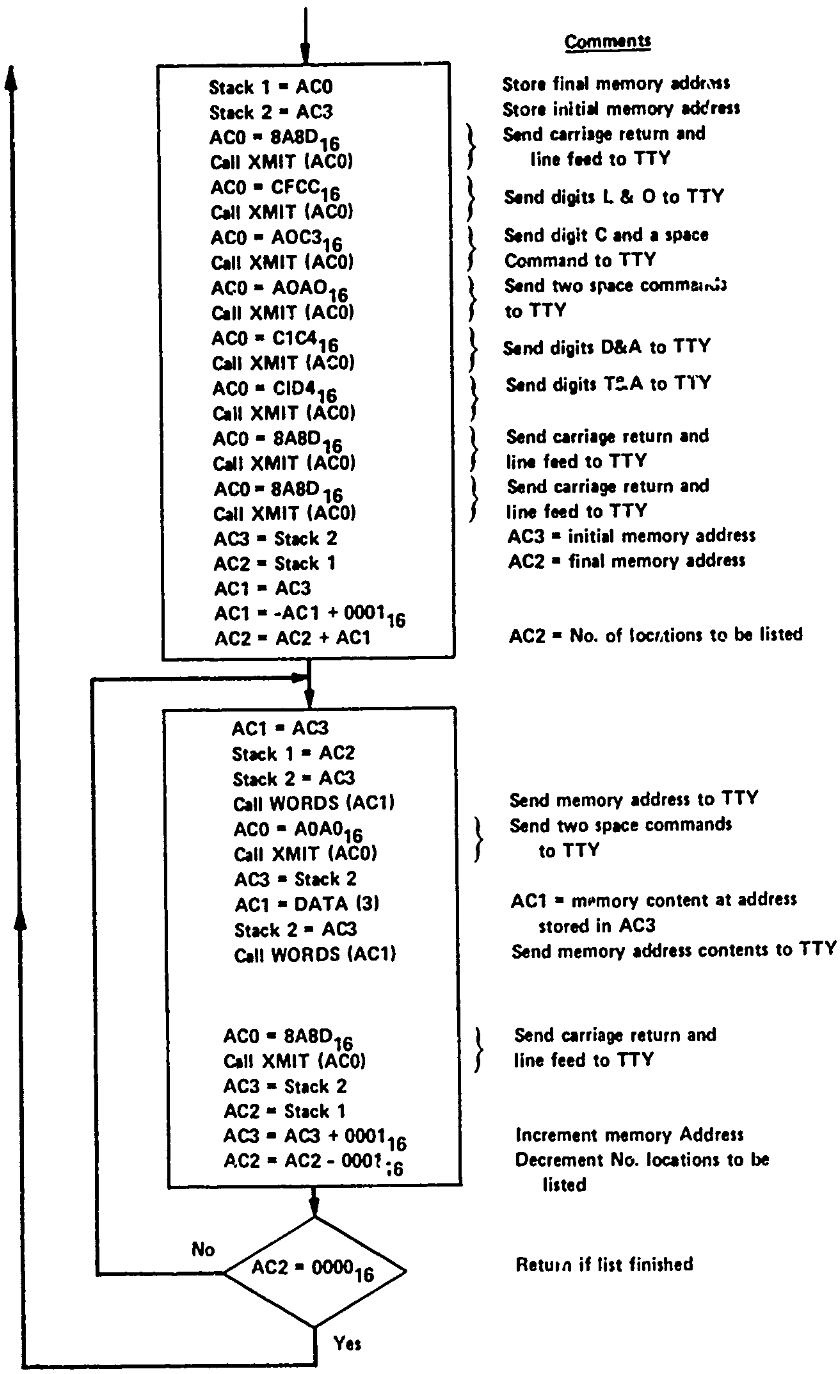

Figure 35. ROM Loader Flow Chart (Cont)

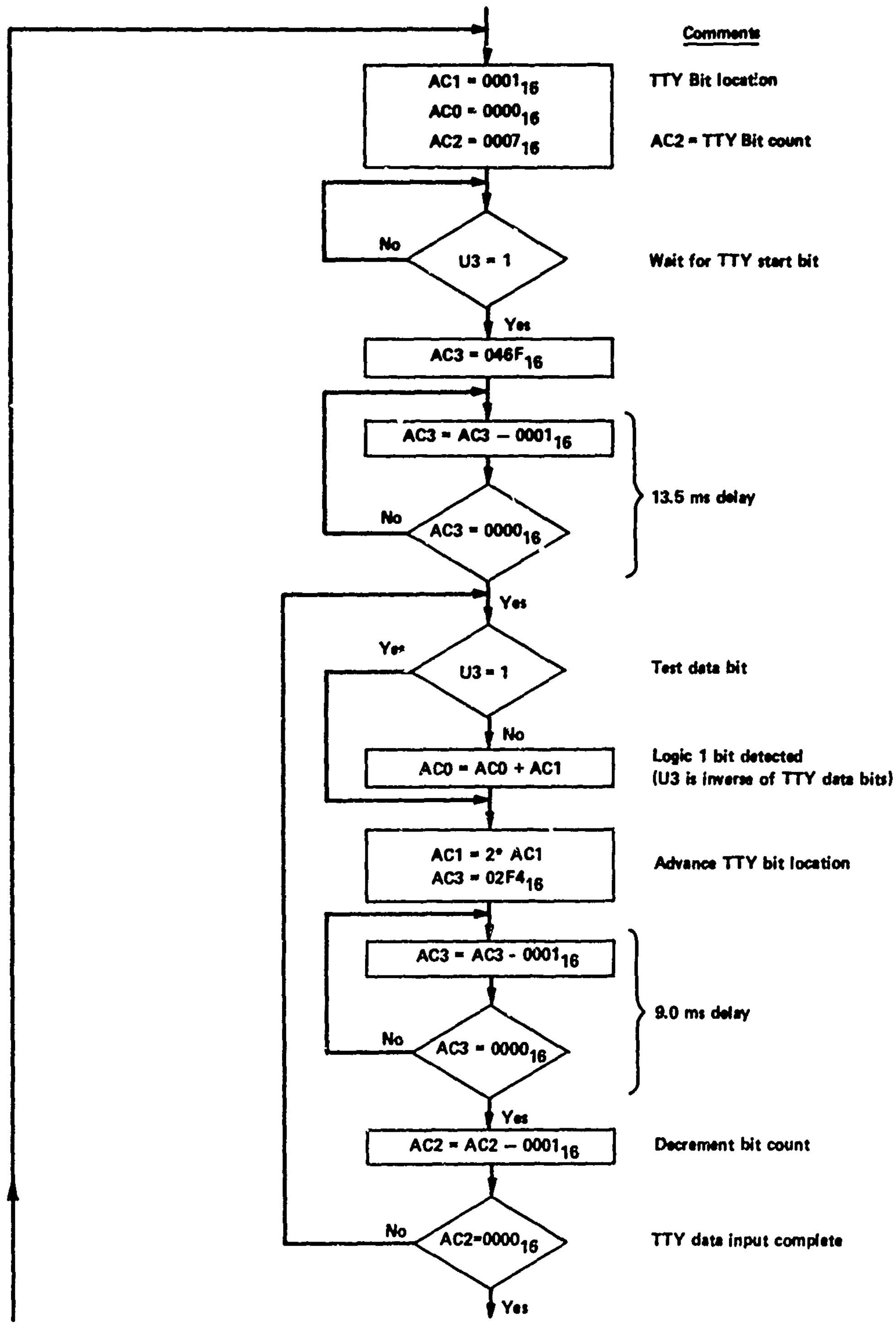

Figure 36. Subroutine RECV (AC0, AC1)

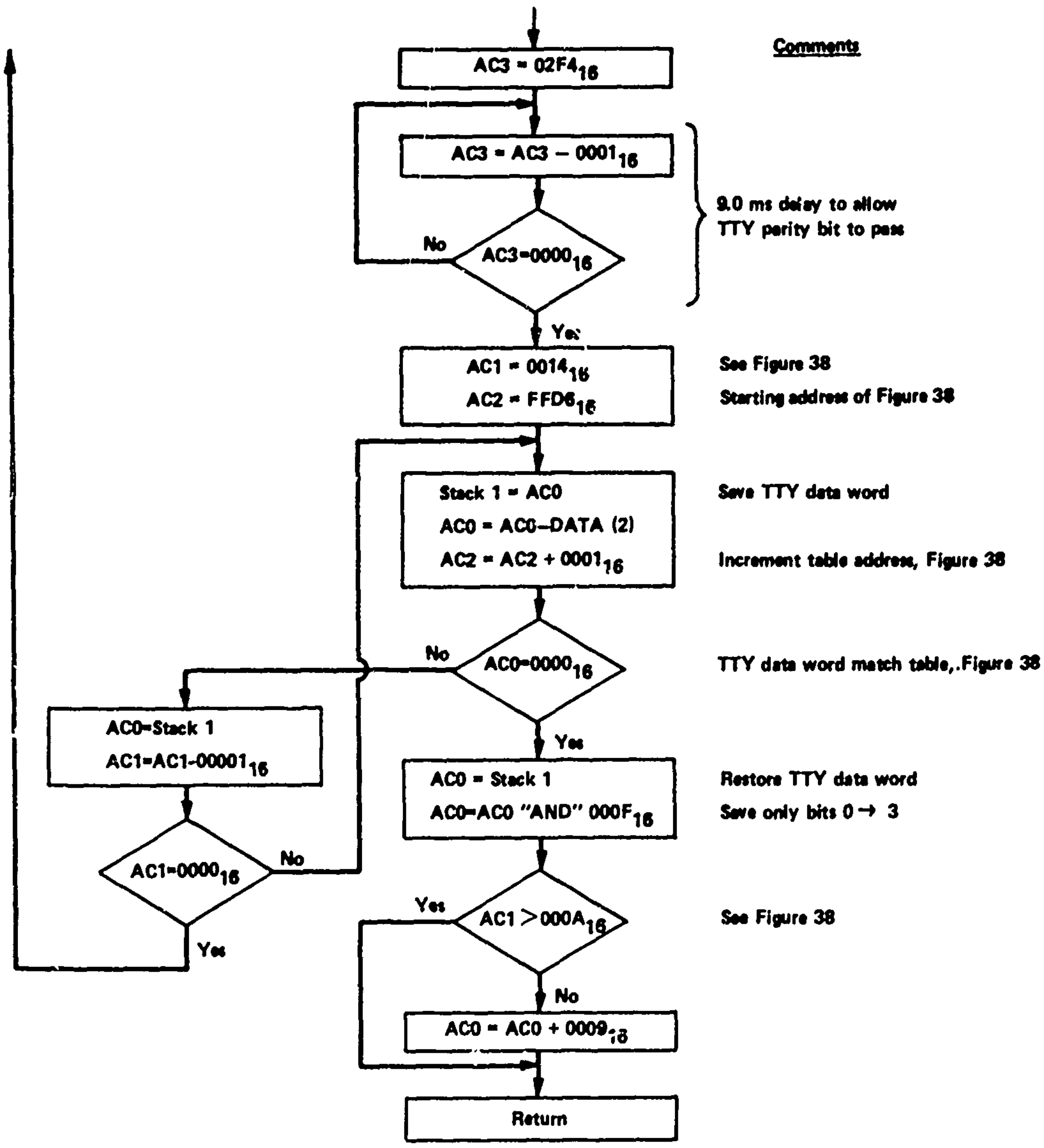

Figure 36. Subroutine RECV (AC0, AC1) (Cont)

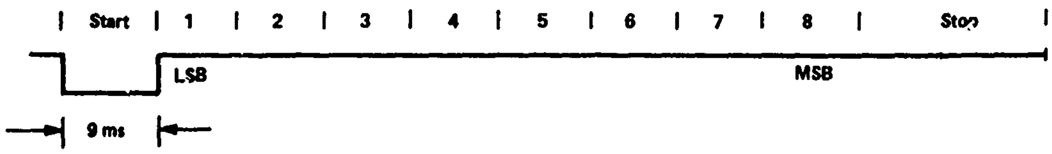

Figure 37. TTY Word Format

DATA(2)	ASCII	AC1
0030_{16}	0	0014_{16}
0031_{16}	1	0013_{16}
0032_{16}	2	0012_{16}
0033_{16}	3	0011_{16}
0034_{16}	4	0010_{16}
0035_{16}	5	$000F_{16}$
0036_{16}	6	$000E_{16}$
0037_{16}	7	$000D_{16}$
0038_{16}	8	$000C_{16}$
0039_{16}	9	$000B_{16}$
0041_{16}	A	$000A_{16}$
0042_{16}	B	0009_{16}
0043_{16}	C	0008_{16}
0044_{16}	D	0007_{16}
0045_{16}	E	0006_{16}
0046_{16}	F	0005_{16}
0021_{16}	!	0004_{16}
$004C_{16}$	L	0003_{16}
0052_{16}	R	0002_{16}
0057_{16}	W	0001_{16}

Figure 38. Subroutine RECV (AC0, AC1) Look Up Table

4.3.2 SUBROUTINE WORDR (AC0, AC1)

Subroutine WORDR (AC0, AC1) receives four data words from the teletype and stores them as four hexadecimal digits of four bits each in AC0 using subroutine RECV (AC0, AC1), (see Figure 39 and Figure 40). AC1 will contain the position in the look up table, Figure 38, of the last data word received. If the teletype sends an !, L, R or W the subroutine terminates early.

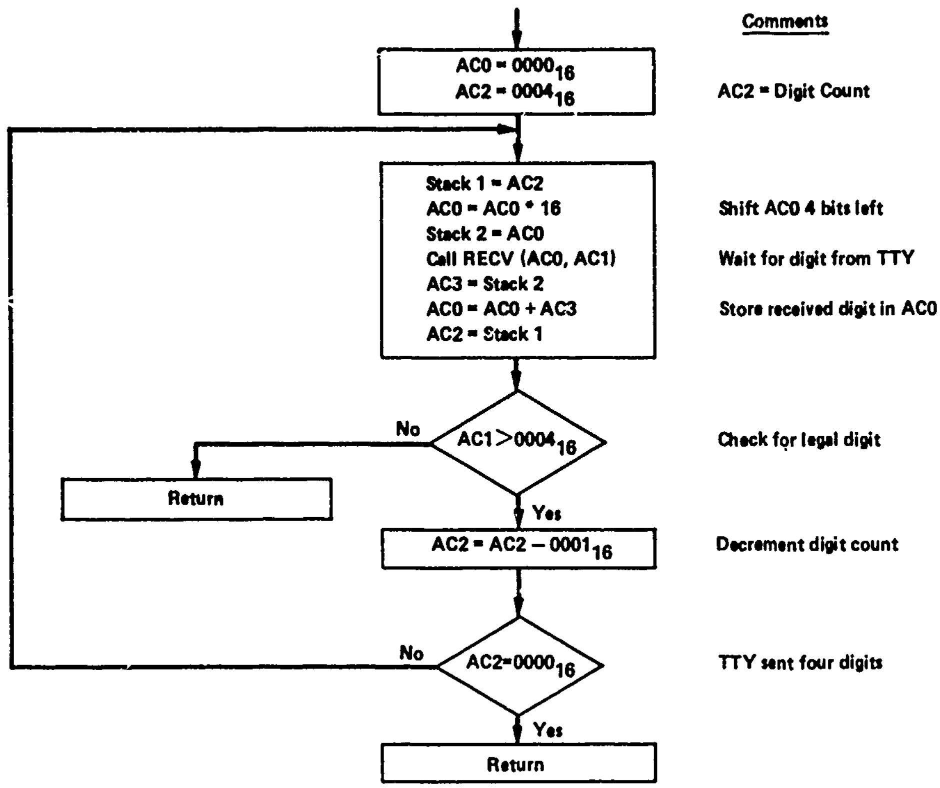

Figure 39. Subroutine WORDR (AC0, AC1)

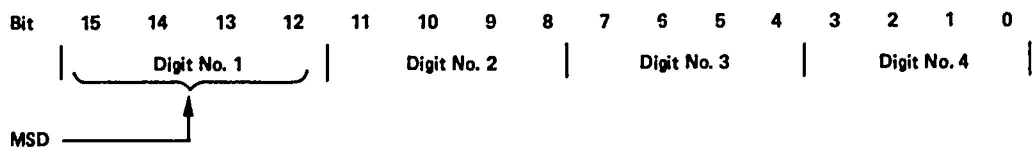

Figure 40. Format of AC0, Subroutine WORDR (AC0, AC1)

4.3.3 SUBROUTINE XMIT (AC0)

Subroutine XMIT (AC0) will transmit two 8 bit data words in AC0 as two teletype data words, (see Figures 41 and 42). If only one data word is to be sent use data word No. 1 and set data word No. 2 to zero in Figure 42.

4.3.4 SUBROUTINE WORDS (AC1)

Subroutine WORDS (AC1) in conjunction with subroutine XMIT (AC0) will transmit to the teletype four hexadecimal digits in AC1 using the format of Figure 40. Figure 43 is the flow chart used by Subroutine WORDS (AC1).

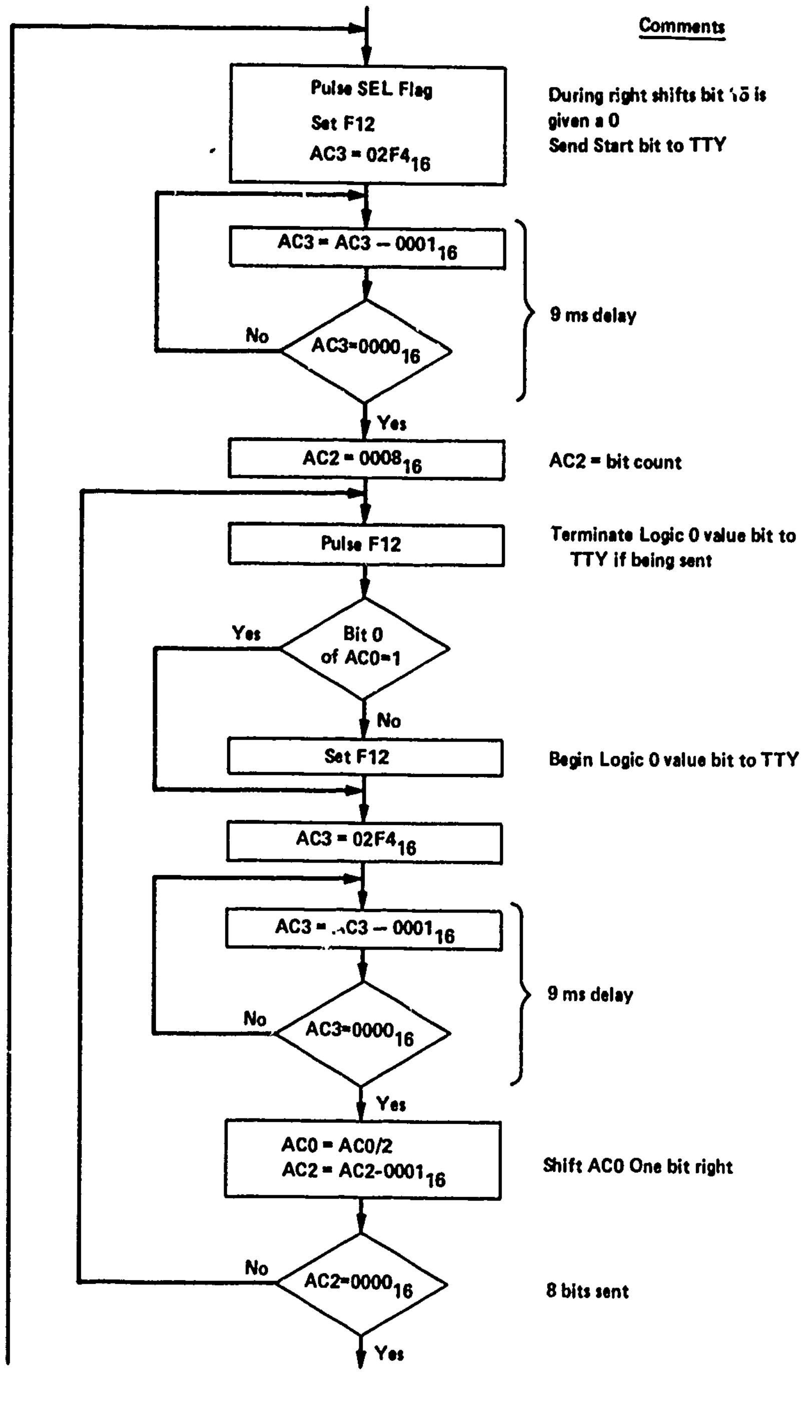

Figure 41. Subroutine XMIT (AC0)

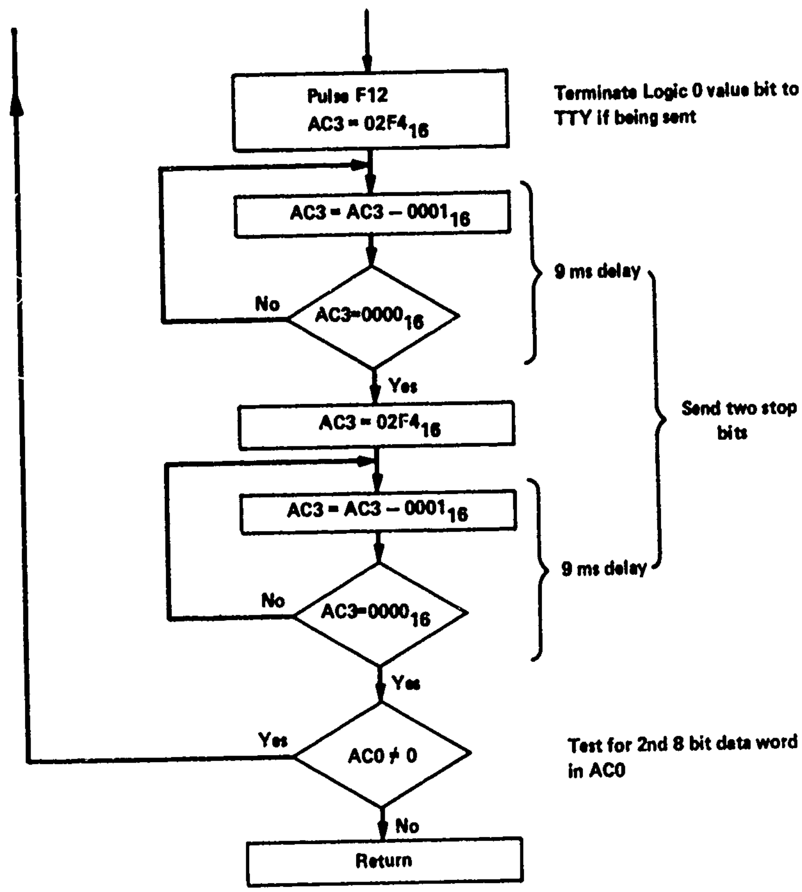

Figure 41. Subroutine XMIT (AC0) (Cont)

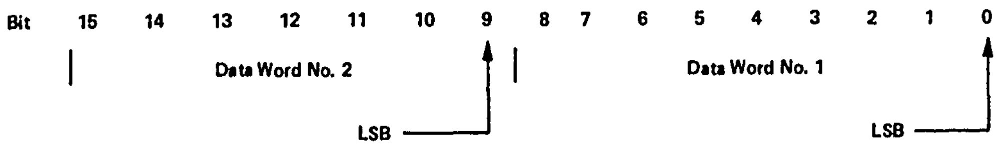

Figure 42. Format of AC0, Subroutine XMIT (AC0)

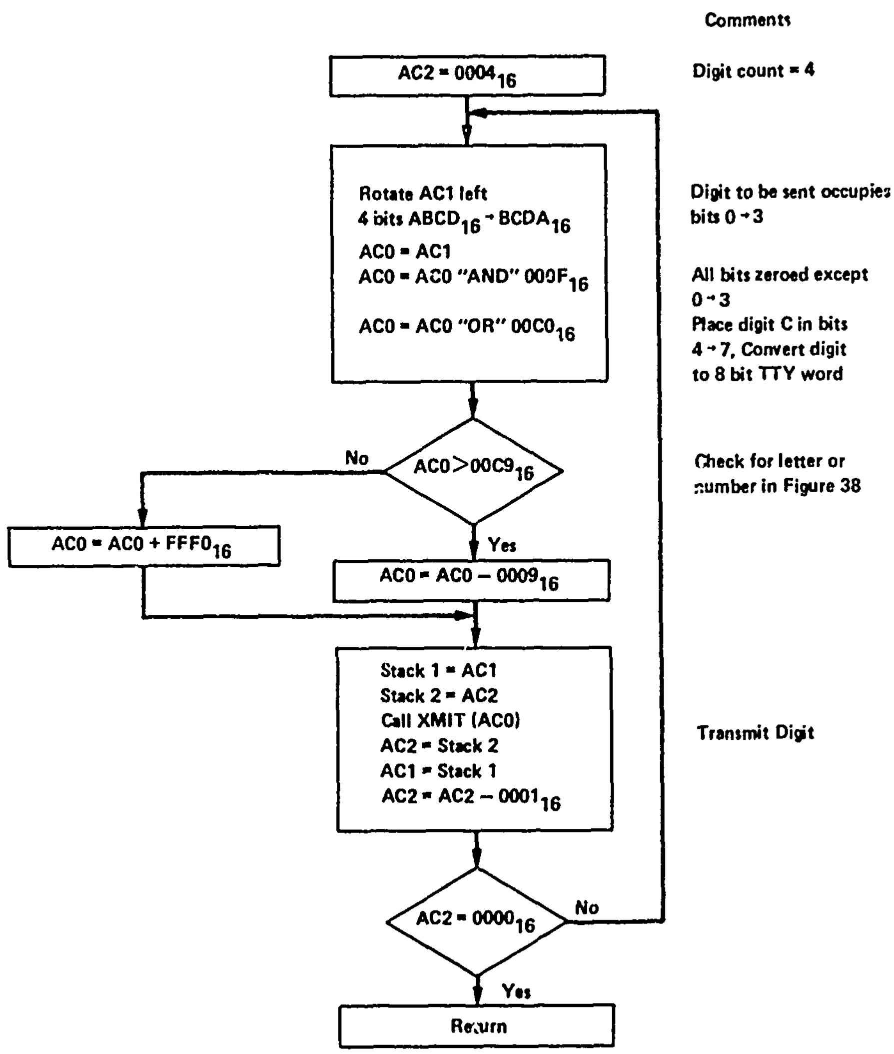

Figure 43. Subroutine WORDS (AC1)

TDY-52B Instruction Set

MNEMONIC	INSTRUCTION NAME	FUNCTION	Execution Time in Microseconds*	Format
LOAD AND STORE				
LD **	LOAD	(EA) → (ACr), IF INDIRECT ((EA)) → (ACr)	9.1, 10.15 IF Ind.	2
ST **	STORE	(ACN) → (EA), IF INDIRECT (ACr) → ((EA))	10.15, 14.35, IF Ind.	2
LDB	LOAD BYTE	(1/2 EA) → (ACO LESS SIGNIFICANT BYTE)	Right 21.0, Left 32.2	5
STB	STORE BYTE	(ACO LESS SIGNIFICANT BYTE → (1/2 EA) 0 → (SEL)	Right 29.05, Left 38.85	5
ARITHMETIC				
ADD	ADD	(ACr) + (EA) → (ACr) OV, CY	9.1	2
SUB	SUBTRACT	(ACr) - (EA) → (ACr) OV, CY	9.1	2
MPY	MULTIPLY	(EA) * (AC1) → {(ACO, (AC1)} L 0 → (SEL)	151.55 to 173.95	5
DIV	DIVIDE	{(ACO), (AC1)} → (EA) → (ACO) QUOTIENT 0 → (SEL) OV, L (AC1) REMAINDER	178.15 to 225.75	5
DADD	DOUBLE PRECISION ADD	{(ACO), (AC1)} + {(EA), (EA+1)} → {(ACO), (AC1)} 0 → (SEL) OV, CY	21.0	5
DSUB	DOUBLE PRECISION SUBTRACT	{(ACO), (AC1)} - {(EA), (EA + 1)} → {(ACO), (AC1)} 0 → (SEL) OV, CY	21.0	5
LOGICAL				
AND	AND	(R01) "AND" (EA) → (R01)	9.1	3
OR	OR	(R01) "OR" (EA) → (R01)	9.1	3
SKIP				
ISZ	INCREMENT AND SKIP IF ZERO	(EA) + 1 → (EA) IF (EA) = 0, (PC) + 1 → (PC)	12.95 If Skip 14.35	4B
DSZ	DECREMENT AND SKIP IF ZERO	(EA) - 1 → (EA) IF (EA) = 0, (PC) + 1 → (PC)	14.35 If Skip 15.75	4B
SKG	SKIP IF GREATER THAN	IF (ACr) > (EA), (PC) + 1 → (PC)	13.3 to 16.1	2
SKNE	SKIP IF NOT EQUAL	IF (ACr) ≠ (EA), (PC) + 1 → (PC)	10.5	2
SKAZ	SKIP IF "AND" IS ZERO	IF (R01) "AND" (EA) = 0, (PC) + 1 → (PC)	10.5 If Skip 11.9	3
SKSTF	SKIP IF STATUS FLAG TRUE	IF (STATUS FLAG N) = 1, (PC) + 1 → (PC) 0 (SEL)	27.65 to 55.65	9
SKBIT	SKIP IF BIT TRUE	IF (ACO BIT N) = 1, (PC) + 1 → (PC) 0 (SEL)	27.65 to 55.65	9
SINGLE BIT				
SETST	SET STATUS BIT	1 → (STATUS FLAG N)	24.85 to 51.45	9
CLRST	CLEAR STATUS BIT	0 → (STATUS FLAG N)	24.85 to 51.45	9
SETBIT	SET BIT	1 → (ACO BIT N)	22.05 to 48.65	9
CLRBIT	CLEAR BIT	0 → (ACO BIT N)	22.05 to 48.65	9
CMPBIT	COMPLEMENT BIT	$\overline{\text{(ACO BIT N)}}$ → (ACO BIT N)	22.05 to 48.65	9
TRANSFER				
JMP **	JUMP	EA → (PC), IF INDIRECT (EA) → (ACr)	5.25 If Ind. 9.1	4A
JSR **	JUMP TO SUBROUTINE	(PC) → (STK) EA → (PC), IF INDIRECT (EA) → (PC)	6.65, If Ind. 10.5	4A
BOC	BRANCH ON CONDITION	IF CONDITION CC IS TRUE, (PC) + D → (PC)	6.65, IF Branch 8.05	1
RTI	RETURN FROM INTERRUPT	(STK) + C → (PC) 1 → (IEF)	8.05	8
RTS	RETURN FROM SUBROUTINE	(STK) + C → (PC)	6.65	8
JSRI	JUMP TO SUBROUTINE IMPLIED	(PC) → (STK) $FF80_{16}$ + C → (PC)	6.65	8

*Times are for T4 Extended 6 Time Phases

**For Ind. add ID to mnemonic, that is, LDID

TDY-52B Instruction Set (Cont)

MNEMONIC	INSTRUCTION NAME	FUNCTION	Execution Time In Microseconds*	Format
TRANSFER (cont)				
JMPP	JUMP THROUGH POINTER	$(100_{16}$ +N) → (PC)	12.95	9
JSRP	JMP TO SUBROUTINE THRU POINTER	(PC) → (STK) $(100_{16}$ + C) → (PC)	14.35	8
INTERRUPT				
JINT	JUMP INDIRECT TO LEVEL 0	(PC) → (STK), 0 → (IEF) $(120_{16}$ +N) → PC	11.9	9
ISCAN	INTERRUPT SCAN	1/2 (AC1) → (AC1) UNTIL 1 SHIFTED OUT (AC2) + NUMBER OF SHIFTS → (AC2)	13.65 to 113.05	9
SHIFT				
ROL	ROTATE LEFT	2 (ACr) → (ACr) IF SEL = 0, (BIT 15) → (BIT 0) IF SEL = 1, (BIT 15) → (L), (L) → (BIT 0) } D TIMES	6.65 + 4.2D	4B
ROR	ROTATE RIGHT	1/2 (ACr) → (ACr) IF SEL = 0, (BIT 0) → (BIT 15) IF SEL = 1, (BIT 0) → (L), (L) → (BIT 15) } D TIMES	6.65 + 4.2D	4B
SHL	SHIFT LEFT	2 (ACr) → (ACr) 0 → (BIT 0) IF SEL = 1, (BIT 15) → (L) } D TIMES	6.65 + 4.2D	4B
SHR	SHIFT RIGHT	1/2 (ACr) → (ACr) IF SEL = 0, 0 → (BIT 15) IF SEL = 1, (L) → (BIT 15), 0 → (L) } D TIMES	6.65 + 4.2D	4B
STACK				
PUSH	PUSH ONTO STACK	(ACr) → (STK)	5.25	4B
PULL	PULL FROM STACK	(STK) → (ACr)	5.25	4B
PUSHF	PUSH STATUS FLAGS ONTO STACK	(SF) → (STK)	6.65	8
PULLF	PULL STATUS FLAGS FROM STACK INTO FLAG REGISTER	(STK) → (ACr)	8.05	8
XCHRS	EXCHANGE REGISTER AND STACK	(ACr) → (STK) (STK) → (ACr)	8.05	4B
IMMEDIATE				
LI	LOAD IMMEDIATE	D → (ACr)	5.25	4B
AISZ	ADD IMMEDIATE AND SKIP IF ZERO	(ACr) + D → (ACr) OV, CY IF (ACr) = 0, (PC) + 1 → (PC)	6.65,IF Skip 8.05	4B
CAI	COMPLEMENT AND ADD IMMEDIATE	~ (ACr) + D → (ACr)	5.25	4B
REGISTER				
RADD	REGISTER ADD	(SR) + (DR) → (DR) OV, CY	5.25	6
RXCH	REGISTER EXCHANGE	(SR) → (DR), (DR) → (SR)	12.25	6
RCPY	REGISTER COPY	(SR) → (DR)	10.5	6
RXOR	REGISTER EXCLUSIVE OR	(SR) ⊙ (DR) → (DR)	9.45	6
RAND	REGISTER AND	(SR) "AND" (DR) → (DR)	9.45	6
INPUT/OUTPUT				
RIN	REGISTER INPUT	(AC3) + C → (IO ADDR) (IO DATA) → (AC0)	10.85	8
ROUT	REGISTER OUTPUT	(AC3) + C → (IOADDR) (AC0) → (IO DATA)	10.85	8
SFLG	SET FLAG	C → (IOADDR), 1 → (CONTROL FLAG FC)	6.65	7
PFLG	PULSE FLAG	C → (IOADDR), 1 → (CONTROL FLAG FC)	6.65	7
HALT	HALT	PROCESSOR HALTS	----	8

*Times are For T4 Extended 6 Time Phases

Appendix B

TDY-52B Assembly Language

Each source statement of Appendices C and D contains an instruction found in Appendix A or a 4 digit hexadecimal data word stored at the indicated memory location. Figure B1 is the instruction format while Figure B2 is the data word format:

Instruction Format

a) Optional address field with not more than ten alphanumeric characters.

b) Instruction mnemonic field. See Appendix A.

c) Working accumulator field or code field with the following formats referenced to Appendix A.
   - Formats 4A, 4B and 5: Working accumulator having one of four values (0, 1, 2, 3) except where restricted by Appendix A in the column labeled function.
   - Format 6: Source accumulator followed by destination accumulator in parenthesis. Each having one of four values (0, 1, 2, 3).
   - Formats 1, 7, and 9: Condition code field containing one of sixteen values (0, 1, ... , 15).
   - Formats 2, 3 and 8: Field not used.

Preceding page blank

d) Displacement field with either ten or less alphanumeric characters referring to an address field, or a 4 digit hexadecimal value whose least significant digits are used to determine the least significant bits of the instructions object code. The addressing mode is placed in parenthesis at the end of the displacement field for those instructions using formats 2, 3, 4A and 5. The addressing mode is one of four values (0, 1, 2, 3).

0 - direct

1 - relative to Program Counter

2 - relative to Accumulator 2

3 - relative to Accumulator 3

Data Format

a) Optional address field with not more than ten alphanumeric characters.

b) Data field having a 4 digit hexadecimal value.

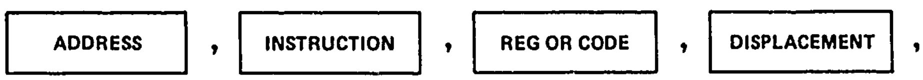

Figure B1. Assembly Language Instruction Format

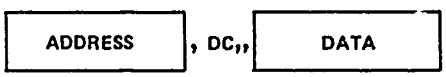

Figure B2. Assembly Language Data Format

AUTOPILOT LISTING

LOC	OBJECT CODE	STMT	SOURCE STATEMENTS
0000	0000	1	,DC,,0000,
0001	0000	2	INT2,DC,,0000,
0002	0000	3	TN,DC,,0000,
0003	0000	4	SCP,DC,,0000,
0004	0000	5	COSPH,DC,,0000,
0005	0000	6	SINPH,DC,,0000,
0006	0000	7	SCP2,DC,,0000,
0007	0000	8	SCP3,DC,,0000,
0008	0000	9	SCP4,DC,,0000,
0009	0000	10	ADPH,DC,,0000,
000A	0000	11	ADPL,DC,,0000,
000B	0000	12	ADPH1,DC,,0000,
000C	0000	13	ADPL1,DC,,0000,
000D	0000	14	ADPH2,DC,,0000,
000E	0000	15	ADPL2,DC,,0000,
000F	0000	16	THETA,DC,,0000,
0010	0000	17	THETA1,DC,,0000,
0011	0000	18	THETA2,DC,,0000,
0012	0000	19	ADYH,DC,,0000,
0013	0000	20	ADYL,DC,,0000,
0014	0000	21	ADYH1,DC,,0000,
0015	0000	22	ADYL1,DC,,0000,
0016	0000	23	ADYH2,DC,,0000,
0017	0000	24	ADYL2,DC,,0000,
0018	0000	25	PSIG,DC,,0000,
0019	0000	26	PSIG1,DC,,0000,
001A	0000	27	PSIG2,DC,,0000,
001B	0000	28	SEGADD,DC,,0000,
001C	0000	29	REFPH,DC,,0000,
001D	0000	30	REFPL,DC,,0000,
001E	0000	31	,DC,,0000,
001F	0000	32	,DC,,0000,
0020	4700	33	,PULL,3,,
0021	4C10	34	,LI,0,0010,
0022	C002	35	,ADD,0,TN(0),
0023	A002	36	,ST,0,TN(0),
0024	0800	37	,SFLG,0,0000,
0025	0900	38	,SFLG,1,0000,
0026	B052	39	,STID,0,COUT(0),
0027	D04C	40	,SUB,0,TL(0),
0028	1205	41	,BOC,2,5,
0029	1E01	42	2,BOC,14,1,
002A	2029	43	,JMP,,2(0),
002B	0880	44	1,PFLG,0,0000,
002C	202D	45	3,JMP,,4(0),
002D	202C	46	4,JMP,,3(0),
002E	8002	47	5,LD,0,TN(0),

Preceding page blank

AUTOPILOT LISTING (cont'd)

LOC	OBJECT CODE	STMT	SOURCE STATEMENTS
002F	D04D	48	,SUB,0,TI(0),
0030	1205	49	,BOC,2,8,
0031	0C00	50	9,SFLG,4,0000,
0032	1E01	51	6,BOC,14,7,
0033	2032	52	,JMP,,6(0),
0034	0C80	53	7,PFLG,4,0000,
0035	202D	54	,JMP,,4(0),
0036	8002	55	8,LD,0,TH(0),
0037	D04E	56	,SUB,0,TI3(0),
0038	1202	57	,BOC,2,10,
0039	0D00	58	,SFLG,5,0000,
003A	2031	59	,JMP,,9(0),
003B	0D80	60	10,PFLG,5,0000,
003C	8002	61	,LD,0,TN(0),
003D	D04F	62	,SUB,0,TF(0),
003E	1B54	63	,BOC,11,TRANS,
003F	8002	64	,LD,0,TN(0),
0040	D050	65	,SUB,0,TF3(0),
0041	1204	66	,BOC,2,11,
0042	4C00	67	,LI,0,0000,
0043	B053	68	,STID,0,POUT(0),
0044	B054	69	,STID,0,YOUT(0),
0045	2031	70	,JMP,,9(0),
0046	0E00	71	11,SFLG,6,0000,
0047	8002	72	,LD,0,TH(0),
0048	D051	73	,SUB,0,T2(0),
0049	1BE2	74	,BOC,11,3,
004A	0F00	75	,SFLG,7,0000,
004B	0000	76	,HALT,,,
004C	0000	77	TL,DC,,0000,
004D	1F20	78	TI,DC,,1F20,
004E	1FC0	79	TI3,DC,,1FC0,
004F	69E0	80	TF,DC,,69E0,
0050	6A80	81	TF3,DC,,6A80,
0051	7C90	82	T2,DC,,7C90,
0052	0FFC	83	COUT,DC,,0FFC,
0053	0BFC	84	POUT,DC,,0BFC,
0054	07FC	85	YOUT,DC,,07FC,
0055	0000	86	SCP1,DC,,0000,
0056	0001	87	SCP5,DC,,0001,
0057	0020	88	THE2,DC,,0020,
0058	7EF9	89	A,DC,,7EF9,
0059	7DF6	90	B,DC,,7DF6,
005A	0161	91	K2,DC,,0161,
005B	3810	92	TG1,DC,,3810,
005C	3EA9	93	REFP1H,DC,,3EA9,
005D	C1A6	94	REFP1L,DC,,C1A6,

LOC	OBJECT CODE	STMT	SOURCE STATEMENTS
005E	FFFF	95	A1TH,DC,,FFFF,
005F	9DC9	96	A1TL,DC,,9DC9,
0060	50F0	97	TG2,DC,,50F0,
0061	3E10	98	REFP2H,DC,,3E10,
0062	D6CF	99	REFP2L,DC,,D6CF,
0063	FFFF	100	A2TH,DC,,FFFF,
0064	8B14	101	A2TL,DC,,8B14,
0065	7FFF	102	TG3,DC,,7FFF,
0066	3D5A	103	REFP3H,DC,,3D5A,
0067	CB6F	104	REFP3L,DC,,CB6F,
0068	FFFF	105	A3TH,DC,,FFFF,
0069	6F04	106	A3TL,DC,,6F04,
006A	7FFF	107	TG4,DC,,7FFF,
006B	0000	108	,DC,,0000,
006C	0000	109	,DC,,0000,
006D	0000	110	,DC,,0000,
006E	0000	111	,DC,,0000,
006F	0000	112	,DC,,0000,
0070	0647	113	,DC,,0000,
0071	0C8B	114	,DC,,0647,
0072	12C8	115	,DC,,0C8B,
0073	18F8	116	,DC,,12C8,
0074	1F19	117	,DC,,18F8,
0075	2528	118	,DC,,1F19,
0076	2B1F	119	,DC,,2528,
0077	30FB	120	,DC,,2B1F,
0078	36BA	121	,DC,,30FB,
0079	3C56	122	,DC,,36BA,
007A	41CE	123	,DC,,3C56,
007B	471C	124	,DC,,41CE,
007C	4C3F	125	,DC,,471C,
007D	5133	126	,DC,,4C3F,
007E	55F5	127	,DC,,5133,
007F	5A82	128	,DC,,55F5,
0080	5ED7	129	,DC,,5A82,
0081	62F2	130	,DC,,5ED7,
0082	66CF	131	,DC,,62F2,
0083	6A6D	132	,DC,,66CF,
0084	6DCA	133	,DC,,6A6D,
0085	70E2	134	,DC,,6DCA,
0086	73B5	135	,DC,,70E2,
0087	7641	136	,DC,,73B5,
0088	7884	137	,DC,,7641,
0089	7A7D	138	,DC,,7884,
008A	7C29	139	,DC,,7A7D,
008B	7D8A	140	,DC,,7C29,
008C	7E9D	141	,DC,,7D8A,
008D	7F62	142	,DC,,7E9D,
008E	7FD8	143	,DC,,7F62,
			,DC,,7FD8,

AUTOPILOT LISTING (Cont'd)

LOC	OBJECT CODE	STMT	SOURCE STATEMENTS
008F	7FFF	144	,DC,,7FFF,
0090	5500	145	CHECK,DC,,5500,
0091	4000	146	YREPH,DC,,4000,
0092	7F00	147	FIN,DC,,7F00,
0093	4F00	148	TRANS,LI,3,0000,
0094	0B00	149	,SFLG,3,0000,
0095	0440	150	,RIN,,0040,
0096	0B80	151	,PFLG,3,0000,
0097	0765	152	,CMPBIT,5,,
0098	0A80	153	,PFLG,2,0000,
0099	5CFB	154	,SHR,0,0005,
009A	3281	155	,RCPY,0(2),,
009B	5EFE	156	,SHR,2,0002,
009C	1408	157	,BOC,4,12
009D	8E6F	158	,LD,3,006F(2),
009E	5201	159	,CAI,2,0001,
009F	C857	160	,ADD,2,THE2(0),
00A0	866F	161	,LD,1,006F(2),
00A1	130A	162	,BOC,3,14,
00A2	5301	163	,CAI,3,0001,
00A3	5101	164	13,CAI,1,0001,
00A4	20AC	165	,JMP,,14(0),
00A5	5241	166	12,CAI,2,0041,
00A6	8E6F	167	,LD,3,006F(2),
00A7	5201	168	,CAI,2,0001,
00A8	C857	169	,ADD,2,THE2(0),
00A9	866F	170	,LD,1,006F(2),
00AA	13F8	171	,BOC,3,13,
00AB	5301	172	,CAI,3,0001,
00AC	A404	173	14,ST,1,COSPH(0),
00AD	AC05	174	,ST,3,SINPH(0),
00AE	8009	175	,LD,0,ADPH(0),
00AF	8404	176	,LD,1,COSPH(0),
00B0	28CB	177	,JSR,,MUL(0),
00B1	A006	178	,ST,0,SCP2(0),
00B2	A407	179	,ST,1,SCP3(0),
00B3	8012	180	,LD,0,ADYH(0),
00B4	8405	181	,LD,1,SINPH(0),
00B5	28CB	182	,JSR,,MUL(0),
00B6	04A0	183	,DADD,,SCP2(0),
00B7	0006		
00B8	28E1	184	,JSR,,AOUT(0),
00B9	A008	185	,ST,0,SCP4(0),
00BA	8009	186	,LD,0,ADPH(0),
00BB	8405	187	,LD,1,SINPH(0),
00BC	28CB	188	,JSR,,MUL(0),
00BD	A006	189	,ST,0,SCP2(0),

AUTOPILOT LISTING (Cont'd)

LOC	OBJECT CODE	STMT	SOURCE STATEMENTS
00BE	A407	190	,ST,1,SCP3(0),
00BF	8012	191	,LD,0,ADYH(0),
00C0	8404	192	,LD,1,COSPH(0),
00C1	28CB	193	,JSR,,MUL(0),
00C2	04B0	194	,DSUB,,SCP2(0),
00C3	0006		
00C4	28E1	195	,JSR,,AOUT(0),
00C5	8408	196	,LD,1,SCP4(0),
00C6	E054	197	,STID,0,YOUT(0),
00C7	B453	198	,STID,1,POUT(0),
00C8	0C00	199	,SFLG,4,0000,
00C9	1E2A	200	16,BOC,14,15,
00CA	20C9	201	,JMP,,16(0),
00CB	1211	202	MUL,BOC,2,100,
00CC	3180	203	,RXCH,0(1),,
00CD	1206	204	,BOC,2,101,
00CE	5101	205	,CAI,1,0001,
00CF	5001	206	,CAI,0,0001,
00D0	A003	207	102,ST,0,SCP(0),
00D1	0480	208	,MPY,,SCP(0),
00D2	0003		
00D3	0200	209	,RTS,,0000,
00D4	5101	210	101,CAI,1,0001,
00D5	A003	211	103,ST,0,SCP(0),
00D6	0480	212	,MPY,,SCP(0),
00D7	0003		
00D8	5000	213	,CAI,0,0000,
00D9	5100	214	,CAI,1,0000,
00DA	04A0	215	,DADD,,SCP1(0),
00DB	0055		
00DC	0200	216	,RTS,,0000,
00DD	3180	217	100,RXCH,0(1),,
00DE	12F1	218	,BOC,2,102,
00DF	5001	219	,CAI,0,0001,
00E0	20D5	220	,JMP,,103(0),
00E1	120E	221	AOUT,BOC,2,18,
00E2	5000	222	,CAI,0,0000,
00E3	5100	223	,CAI,1,0000,
00E4	04A0	224	,DADD,,SCP1(0),
00E5	0055		
00E6	4F01	225	,LI,3,0001,
00E7	E0F2	226	19,SKG,0,DSA(0),
00E8	20EE	227	,JMP,,17(0),
00E9	8092	228	,LD,0,FIN(0),
00EA	FC56	229	20,SKNE,3,SCP5(0),
00EB	0200	230	,RTS,,,
00EC	68F3	231	,OR,0,PFIN(0),
00ED	0200	232	,RTS,,,
00EE	5C07	233	17,SHL,0,0007,
00EF	20EA	234	,JMP,,20(0),
00F0	4F00	235	18,LI,3,0000,

AUTOPILOT LISTING (Cont'd)

LOC	OBJECT CODE	STMT	SOURCE STATEMENTS
00F1	20E7	236	,JMP,,19(0),
00F2	00FF	237	DSA,DC,,00FF,
00F3	8000	238	PFIN,DC,,8000,
00F4	4F00	239	15,LI,3,0000,
00F5	0B00	240	,SFLG,3,0000,
00F6	0400	241	,RIN,,0000,
00F7	0B80	242	,PFLG,3,0000,
00F8	A018	243	,ST,0,PSIG(0),
00F9	0B00	244	,SFLG,3,0000,
00FA	0420	245	,RIN,,0020,
00FB	0B80	246	,PFLG,3,0000,
00FC	A00F	247	,ST,0,THETA(0),
00FD	881B	248	,LD,2,SEGADD(0),
00FE	8200	249	,LD,0,0000(2),
00FF	D002	250	,SUB,0,TH(0),
0100	1B14	251	,BOC,11,500,
0101	801C	252	501,LD,0,REFPH(0),
0102	D00F	253	,SUB,0,THETA(0),
0103	5C01	254	,SHL,0,0001,
0104	A00F	255	,ST,0,THETA(0),
0105	801C	256	,LD,0,REFPH(0),
0106	841D	257	,LD,1,REFPL(0),
0107	06A0	258	,DADD,,0003(2),
0108	0003		
0109	A01C	259	,ST,0,REFPH(0),
010A	A41D	260	,ST,1,REFPL(0),
010B	4F09	261	,LI,3,0009,
010C	290F	262	,JSR,,CALC(1),
010D	8091	263	,LD,0,YREPH(0),
010E	D018	264	,SUB,0,PSIG(0),
010F	5C01	265	,SHL,0,0001,
0110	A018	266	,ST,0,PSIG(0),
0111	4F12	267	,LI,3,0012,
0112	2909	268	,JSR,,CALC(1),
0113	0C80	269	,PFLG,4,0000,
0114	202C	270	,JMP,,3(0),
0115	4A05	271	500,AISZ,2,0005,
0116	A81B	272	,ST,2,SEGADD(0),
0117	8201	273	,LD,0,0001(2),
0118	A01C	274	,ST,0,REFPH(0),
0119	8202	275	,LD,0,0002(2),
011A	A01D	276	,ST,0,REFPL(0),
011B	21E5	277	,JMP,,501(1),
011C	8308	278	CALC,LD,0,0008(3),
011D	845A	279	,LD,1,K2(0),
011E	28CB	280	,JSR,,MUL(0),
011F	4E00	281	,LI,2,0000,
0120	1201	282	,BOC,2,200,
0121	4E80	283	,LI,2,0080,
0122	4300	284	200,PUSH,3,,

AUTOPILOT LISTING (Cont'd)

LOC	OBJECT CODE	STMT	SOURCE STATEMENTS
0123	CA00	285	,SFLG,2,0000,
0124	4F08	286	,LI,3,0008,
0125	4200	287	201,PUSH,2,,
0126	0280	288	,PULLF,,,
0127	58FF	289	,ROR,0,0001,
0128	59FF	290	,ROR,1,0001,
0129	4BFF	291	,AISZ,3,00FF,
012A	21FA	292	,JMP,,201(1),
012B	4700	293	,PULL,3,,
012C	4000	294	,PUSH,0,,
012D	4100	295	,PUSH,1,,
012E	8304	296	,LD,0,0004(3),
012F	8705	297	,LD,1,0005(3),
0130	120A	298	,BOC,2,600,
0131	5000	299	,CAI,0,0000,
0132	5100	300	,CAI,1,0000,
0133	04A0	301	,DADD,,SCP1,
0134	0055		
0135	2935	302	,JSR,,MPYB(1),
0136	5000	303	,CAI,0,0000,
0137	5100	304	,CAI,1,0000,
0138	04A0	305	,DADD,,SCP1,
0139	0055		
013A	2101	306	,JMP,,601(1),
013B	292F	307	600,JSR,,MPYB(1),
013C	A006	308	601,ST,0,SCP2(0),
013D	A407	309	,ST,1,SCP3(0),
013E	4500	310	,PULL,1,,
013F	4400	311	,PULL,0,,
0140	04B0	312	,DSUB,,SCP2(0),
0141	0006		
0142	4000	313	,PUSH,0,,
0143	4100	314	,PUSH,1,,
0144	8302	315	,LD,0,0002(3),
0145	8703	316	,LD,1,0003(3),
0146	120A	317	,BOC,2,700,
0147	5000	318	,CAI,0,0000,
0148	5100	319	,CAI,1,0000,
0149	04A0	320	,DADD,,SCP1,
014A	0055		
014B	2946	321	,JSR,,MPYA(1),
014C	5000	322	,CAI,0,0000,
014D	5100	323	,CAI,1,0000,
014E	04A0	324	,DADD,,SCP1,
014F	0055		

AUTOPILOT LISTING (Cont'd)

LOC	OBJECT CODE	STMT	SOURCE STATEMENTS
0150	2101	325	,JMP,,701(1),
0151	2940	326	700,JSR,,MPYA(1),
0152	0A00	327	701,SFLG,2,0000,
0153	5D01	328	,SHL,1,0001,
0154	5801	329	,ROL,0,0001,
0155	A006	330	,ST,0,SCP2(0),
0156	A407	331	,ST,1,SCP3(0),
0157	4500	332	,PULL,1,,
0158	4400	333	,PULL,0,,
0159	04A0	334	,DADD,,SCP2(0),
015A	0006		
015B	8E07	335	,LD,2,0007(3),
015C	AB08	336	,ST,2,0008(3),
015D	8B06	337	,LD,2,0006(3),
015E	AB07	338	,ST,2,0007(3),
015F	8B03	339	,LD,2,0003(3),
0160	AB05	340	,ST,2,0005(3),
0161	8B02	341	,LD,2,0002(3),
0162	AB04	342	,ST,2,0004(3),
0163	A300	343	,ST,0,0000(3),
0164	A701	344	,ST,1,0001(3),
0165	0A00	345	,SFLG,2,0000,
0166	5D01	346	,SHL,1,0001,
0167	5801	347	,ROL,0,0001,
0168	A302	348	,ST,0,0002(3),
0169	A703	349	,ST,1,0003(3),
016A	0200	350	,RTS,,,
016B	4000	351	MPYB,PUSH,0,,
016C	4100	352	,PUSH,1,,
016D	8459	353	,LD,1,B(0),
016E	A003	354	,ST,0,SCP(0),
016F	0480	355	,MPY,,SCP(0),
0170	0003		
0171	A006	356	,ST,0,SCP2(0),
0172	A407	357	,ST,1,SCP3(0),
0173	4400	358	,PULL,0,,
0174	0A80	359	,PFLG,2,0000,
0175	5CFF	360	,SHR,0,0001,
0176	8459	361	,LD,1,B(0),
0177	A003	362	,ST,0,SCP(0),
0178	0480	363	,MPY,,SCP(0),
0179	0003		

AUTOPILOT LISTING (Cont'd)

LOC	OBJECT CODE	STMT	SOURCE STATEMENTS
017A	0A00	364	,SFLG,2,0000,
017B	5D01	365	,SHL,1,0001,
017C	5801	366	,ROL,0,0001,
017D	3181	367	,RCPY,0(1),,
017E	4C00	368	,LI,0,0000,
017F	04A0	369	,DADD,,SCP2,
0180	0006		
0181	A006	370	,ST,0,SCP2(0),
0182	A407	371	,ST,1,SCP3(0),
0183	8136	372	,LD,0,BL(1),
0184	0A80	373	,PFLG,2,0000,
0185	5CFF	374	,SHR,0,0001,
0186	4500	375	,PULL,1,,
0187	A003	376	,ST,0,SCP(0),
0188	A480	377	,MPY,,SCP(0),
0189	0003		
018A	0A00	378	,SFLG,2,0000,
018B	5D01	379	,SHL,1,0001,
018C	5801	380	,ROL,0,0001,
018D	3181	381	,RCPY,0(1),,
018E	4C00	382	,LI,0,0000,
018F	04A0	383	,DADD,,SCP2(0),
0190	0006		
0191	0200	384	,RTS,,,
0192	4000	385	MPYA,PUSH,0,,
0193	4100	386	,PUSH,1,,
0194	8458	387	,LD,1,A(0),
0195	A003	388	,ST,0,SCP(0),
0196	0480	389	,MPY,,SCP(0),
0197	0003		
0198	A006	390	,ST,0,SCP2(0),
0199	A407	391	,ST,1,SCP3(0),
019A	4400	392	,PULL,0,,
019B	0A80	393	,PFLG,2,0000,
019C	5CFF	394	,SHR,0,0001,
019D	8458	395	,LD,1,A(0),
019E	A003	396	,ST,0,SCP(0),
019F	0480	397	,MPY,,SCP(0),
01A0	0003		
01A1	0A00	398	,SFLG,2,0000,
01A2	5D01	399	,SHL,1,0001,
01A3	5801	400	,ROL,0,0001,
01A4	3181	401	,RCPY,0(1),,
01A5	4C00	402	,LI,0,0000,
01A6	04A0	403	,DADD,,SCP2,
01A7	0006		

AUTOPILOT LISTING (Cont'd)

LOC	OBJECT CODE	STMT	SOURCE STATEMENTS
01A8	A006	404	,ST,0,SCP2(0),
01A9	A407	405	,ST,1,SCP3(0),
01AA	810E	406	,LD,0,AL(1),
01AB	0A80	407	,PFLG,2,0000,
01AC	5CFF	408	,SHR,0,0001,
01AD	4500	409	,PULL,1,,
01AE	A003	410	,ST,0,SCP(0),
01AF	0480	411	,MPY,,SCP(0),
01B0	0003		
01B1	0A00	412	,SFLG,2,0000,
01B2	5D01	413	,SHL,1,0001,
01B3	5801	414	,ROL,0,0001,
01B4	3181	415	,RCPY,0(1),,
01B5	4C00	416	,LI,0,0000,
01B6	04A0	417	,DADD,,SCP2(0),
01B7	0006		
01B8	0200	418	,RTS,,,
01B9	F902	419	AL,DC,,F902,
01BA	0A68	420	BL,DC,,0A68,
01BB	8120	421	INTT,LD,0,PTEN(1),
01BC	2924	422	,JSR,,FDELAY(1),
01BD	811F	423	,LD,0,PFIVE(1),
01BE	2922	424	,JSR,,FDELAY(1),
01BF	811F	425	,LD,0,NFIVE(1),
01C0	2920	426	,JSR,,FDELAY(1),
01C1	811C	427	,LD,0,NTEN(1),
01C2	291E	428	,JSR,,FDELAY(1),
01C3	8117	429	,LD,0,INT1(1),
01C4	A001	430	,ST,0,INT2(0),
01C5	4C00	431	,LI,0.0000,
01C6	A002	432	,ST,0,TN(0),
01C7	4F12	433	,LI,3,0012,
01C8	4E09	434	,LI,2,0009,
01C9	A200	435	605,ST,0,0000(2),
01CA	4A01	436	,AISZ,2,0001,
01CB	4BFF	437	,AISZ,3,00FF,
01CC	21FC	438	,JMP,,605(1),
01CD	4C5B	439	,LI,0,005B,
01CE	A01B	440	,ST,0,SEGADD(0),
01CF	805C	441	,LD,0,REFP1H(0),
01D0	A01C	442	,ST,0,REFPH(0),
01D1	805D	443	,LD,0,REFP1L(0),
01D2	A01D	444	,ST,0,REFPL(0),
01D3	4C00	445	,LI,0,0000,

AUTOPILOT LISTING (Cont'd)

LOC	OBJECT CODE	STMT	SOURCE STATEMENTS
01D4	B053	446	,STID,0,POUT(0),
01D5	B054	447	,STID,0,YOUT(0),
01D6	B052	448	,STID,0,COUT(0),
01D7	8090	449	,LD,0,CHECK,
01D8	0900	450	,SFLG,1,0000,
01D9	B052	451	,STID,0,COUT(0),
01DA	202C	452	,JMP,,3(0),
01DB	2020	453	INT1,DC,,2020,
01DC	FF00	454	PTEN,DC,,FF00,
01DD	C000	455	PFIVE,DC,,C000,
01DE	7F00	456	NTEN,DC,,7F00,
01DF	4000	457	NFIVE,DC,,4000,
01E0	753C	458	DELAY,DC,,753C,
01E1	B053	459	FDELAY,STID,0,POUT(0),
01E2	B054	460	,STID,0,YOUT(0),
01E3	4D0E	461	,LI,1,000E,
01E4	81FB	462	900,LD,0,DELAY(1),
01E5	48FF	463	901,AISZ,0,00FF,
01E6	21FE	464	,JMP,,901(1),
01E7	49FF	465	,AISZ,1,00FF,
01E8	21FB	466	,JMP,,900(1),
01E9	0200	467	,RTS,,,
01EA	21D0	468	,JMP,,INT1(1),
01EB	21CF	469	,JMP,,INT1(1),
.	.	.	
.	.	.	
01FE	2500	470	,JMPID,,INTI1(1),
01FF	01EA	471	INTI1,DC,,01EA

ROM LOADER LISTING

LOC	OBJECT CODE	STMT	SOURCE STATEMENTS
FF13	0F80	20	TTY,PFLG,7,0000,
FF14	8182	21	,LD,0,LFCR(1),
FF15	295A	22	,JSR,,XMIT(1),
FF16	8181	23	,LD,0,OC(1),
FF17	2958	24	,JSR,,XMIT(1),
FF18	8180	25	,LD,0,MM(1),
FF19	2956	26	,JSR,,XMIT(1),
FF1A	817F	27	,LD,0,NA(1),
FF1B	2954	28	,JSR,,XMIT(1),
FF1C	817E	29	,LD,0,SICD(1),
FF1D	2952	30	,JSR,,XMIT(1),
FF1E	2D50	31	,JSRID,,50(1),
FF1F	E562	32	,SKG,1,FOUR(1),
FF20	2101	33	,JMP,,11(1),
FF21	21F1	34	,JMP,,TTY(1),
FF22	E55D	35	11,SKG,1,ONE(1),
FF23	2140	36	,JMP,,WRITE(1),
FF24	E55C	37	,SKG,1,TWO(1),
FF25	2131	38	,JMP,,READ(1),
FF26	2962	39	LIST,JSR,,WORDR(1),
FF27	E55A	40	,SKG,1,FOUR(1),
FF28	21EA	41	,JMP,,TTY(1),
FF29	4000	42	,PUSH,0,,
FF2A	295E	43	,JSR,,WORDR(1),
FF2B	4700	44	,PULL,3,,
FF2C	E555	45	,SKG,1,FOUR(1),
FF2D	21E5	46	,JMP,,TTY(1),
FF2E	4000	47	,PUSH,0,,
FF2F	4300	48	,PUSH,3,,
FF30	8166	49	,LD,0,LFCR(1),
FF31	293E	50	,JSR,,XMIT(1),
FF32	8169	51	,LD,0,OL(1),
FF33	293C	52	,JSR,,XMIT(1),
FF34	8168	53	,LD,0,SPC(1),
FF35	293A	54	,JSR,,XMIT(1),
FF36	8167	55	,LD,0,SPSP(1),
FF37	2938	56	,JSR,,XMIT( ),
FF38	8166	57	,LD,0,AD(1`,
FF39	2936	58	,JSR,,XMIT(1),
FF3A	8165	59	,LD,0,AF(1),
FF3B	2934	60	,JSR,,XMIT(1),
FF3C	815A	61	,LD,0,LFCR(1),
FF3D	2932	62	,JSR,,XMIT(1),
FF3E	8158	63	,LD,0,LFCR(1),
FF3F	2930	64	,JSR,,XMIT(1),

ROM LOADER LISTING (Cont)

LOC	OBJECT CODE	STMT	SOURCE STATEMENTS
FF40	4700	65	,PULL,3,,
FF41	4600	66	,PULL,2,,
FF42	3D31	67	,RCPY,3,(1),
FF43	5102	68	,CAI,1,0002,
FF44	3600	69	,RADD,1(2),,
FF45	3D81	70	16,RCPY,3(1),,
FF46	4200	71	,PUSH,2,,
FF47	4300	72	,PUSH,3,,
FF48	2958	73	,JSR,,WORDS(1),
FF49	8154	74	,LD,0,SPSP(1),
FF4A	2925	75	,JSR,,XMIT(1),
FF4B	4700	76	,PULL,3,,
FF4C	8700	77	,LD,1,0000(3),
FF4D	4300	78	,PUSH,3,,
FF4E	2952	79	,JSR,,WORDS(1),
FF4F	8147	80	,LD,0,LFCR(1),
FF50	291F	81	,JSR,,XMIT(1),
FF51	4700	82	,PULL,3,,
FF52	4600	83	,PULL,2,,
FF53	CD2C	84	,ADD,3,ONE(1),
FF54	4AFF	85	,AISZ,2,00FF,
FF55	21EF	86	,JMP,,16(1),
FF56	21BC	87	,JMP,,TTY(1),
FF57	0F00	88	READ,SFLG,7,0000,
FF58	2930	89	,JSR,,WORDR(1),
FF59	3381	90	,RCPY,0(3),,
FF5A	4300	91	30,PUSH,3,,
FF5B	292D	92	,JSR,,WORDR(1),
FF5C	4700	93	,PULL,3,,
FF5D	E524	94	,SKG,1,FOUR(1),
FF5E	2103	95	,JMP,,31(1),
FF5F	A300	96	,ST,0,0000(3),
FF60	CD1F	97	,ADD,3,ONE(1),
FF61	21F8	98	,JMP,,30(1),
FF62	0F80	99	31,PFLG,7,0000,
FF63	21AF	100	,JMP,,TTY(1),
FF64	2924	101	WRITE,JSR,,WORDR(1),
FF65	E51C	102	,SKG,1,FOUR(1),
FF66	21AC	103	,JMP,,TTY(1),
FF67	4000	104	,PUSH,0,,
FF68	2920	105	,JSR,,WORDR(1),
FF69	4700	106	,PULL,3,,
FF6A	E517	107	,SKG,1,FOUR(1),
FF6B	21A7	108	,JMP,,TTY(1),
FF6C	A300	109	,ST,0,0000(3),
FF6D	21A5	110	,JMP,,TTY(1),
FF6E	0000	111	.DC,,0000,

ROM LOADER LISTING (Cont)

LOC	OBJECT CODE	STMT	SOURCE STATEMENTS
FF6F	FFB8	112	50,DC,,FFB8,
FF70	0A80	113	XMIT,PFLG,2,0000,
FF71	0C00	114	,SLFG,4,0000,
FF72	2910	115	,JSR,,DELAY(1),
FF73	4E08	116	,LI,2,0008,
FF74	0C80	117	4,PFLG,4,0000,
FF75	1301	118	,BOC,3,3,
FF76	0C00	119	,SFLG,4,0000,
FF77	290B	120	3,JSR,,DELAY(1),
FF78	5CFF	121	,SHR,0,0001,
FF79	4AFF	122	,AISZ,2,00FF,
FF7A	21F9	123	,JMP,,4(1),
FF7B	0C80	124	,PFLG,4,0000,
FF7C	2906	125	,JSR,,DELAY(1),
FF7D	2905	126	,JSR,,DELAY(1),
FF7E	15F1	127	,BOC,5,XMIT,
FF7F	0200	128	,RTS,,,
FF80	0001	129	ONE,DC,,0001,
FF81	0002	130	TWO,DC,,0002,
FF82	0004	131	FOUR,DC,,0004,
FF83	8D04	132	DELAY,LD,3,V2(1),
FF84	4BFF	133	2,AISZ,3,00FF,
FF85	21FE	134	,JMP,,2(1),
FF86	0200	135	,RTS,,,
FF87	046F	136	V1,DC,,046F,
FF88	02F4	137	V2,DC,,02F4,
FF89	4C00	138	WORDR,LI,0,0000,
FF8A	4E04	139	,LI,2,0004,
FF8B	4200	140	20,PUSH,2,,
FF8C	5C04	141	,SHL,0,0004,
FF8D	4000	142	,PUSH,0,,
FF8E	2929	143	,JSR,,RECV(1),
FF8F	4700	144	,PULL,3,,
FF90	3C00	145	,RADD,3(0),,
FF91	4600	146	,PULL,2,,
FF92	E5EF	147	,SKG,1,FOUR(1),
FF93	0200	148	,RTS,,,
FF94	4AFF	149	,AISZ,2,00FF,
FF95	21F5	150	,JMP,,20(1),
FF96	0200	151	,RTS,,,
FF97	8A8D	152	LFCR,DC,,8A8D,
FF98	CFC3	153	OC,DC,,CFC3,
FF99	CDCD	154	MM,DC,,CDCD,
FF9A	CEC1	155	NA,DC,,CEC1,
FF9B	BAC4	156	SICD,DC,,BAC4,
FF9C	CFCC	157	OL,DC,,CFCC,
FF9D	A0C3	158	SPC,DC,,A0C3,
FF9E	A0A0	159	SPSP,DC,,A0A0,

ROM LOADER LISTING (Cont)

LOC	OBJECT CODE	STMT	SOURCE STATEMENTS
FF9F	C1C4	160	AD,DC,,C1C4,
FFA0	C1D4	161	AT,DC,,C1D4,
FFA1	4E04	162	WORDS,LI,2,0004,
FFA2	0A80	163	17,PFLG,2,0000,
FFA3	5904	164	,ROL,1,0004,
FFA4	3481	165	,RCPY,1(0),,
FFA5	610E	166	,AND,0,BLANK(1),
FFA6	690E	167	,OR,0,C(1),
FFA7	E10E	168	,SKG,0,C9(1),
FFA8	2109	169	,JMP,,300(1),
FFA9	D142	170	,SUB,0,NINE(1),
FFAA	4100	171	301,PUSH,1,,
FFAB	4200	172	,PUSH,2,,
FFAC	29C3	173	,JSR,,XMIT(1),
FFAD	4600	174	,PULL,2,,
FFAE	4500	175	,PULL,1,,
FFAF	4AFF	176	,AISZ,2,00FF,
FFB0	21F1	177	,JMP,,17(1),
FFB1	0200	178	,RTS,,,
FFB2	C104	179	300,ADD,0,B(1),
FFB3	21F6	180	,JMP,,301(1),
FFB4	000F	181	BLANK,DC,,000F,
FFB5	00C0	182	C,DC,,00C0,
FFB6	00C9	183	C9,DC,,00C9,
FFB7	FFF0	184	B,DC,,FFF0,
FFB8	4D01	185	RECV,LI,1,0001,
FFB9	4C00	186	,LI,0,0000,
FFBA	4E07	187	,LI,2,0007,
FFBB	1E01	188	6,BOC,14,5,
FFBC	21FE	189	,JMP,,6(1),
FFBD	8DC9	190	5,LD,3,V1(1),
FFBE	29C5	191	,JSR,,2(1),
FFBF	1E01	192	8,BOC,14,7,
FFC0	3400	193	,RADD,1(0),,
FFC1	5D01	194	7,SHL,1,0001,
FFC2	29C0	195	,JSR,,DELAY(1),
FFC3	4AFF	196	,AISZ,2,00FF,
FFC4	21FA	197	,JMP,,8(1),
FFC5	29BD	198	,JSR,,DELAY(1),
FFC6	4D14	199	,LI,1,0014,
FFC7	890D	200	,LD,2,DATA(1),
FFC8	4000	201	9,PUSH,0,,
FFC9	D200	202	,SUB,0,0000(2),
FFCA	C9B5	203	,ADD,2,ONE(1),
FFCB	1104	204	,BOC,1,10,

LOC	OBJECT CODE	STMT	SOURCE STATEMENTS
FFCC	4400	205	,PULL,0,,
FFCD	49FF	206	,AISZ,1,00FF,
FFCE	21F9	207	,JMP,,9(1),
FFCF	21E8	208	,JMP,,RECV(1),
FFD0	4400	209	10,PULL,0,,
FFD1	6118	210	,AND,0,TR(1),
FFD2	E518	211	,SKG,1,TEN(1),
FFD3	C118	212	,ADD,0,NINE(1),
FFD4	0200	213	,RTS,,,
FFD5	FFD6	214	DATA,DC,,FFD6,
FFD6	0030	215	,DC,,0030,
FFD7	0031	216	,DC,,0031,
FFD8	0032	217	,DC,,0032,
FFD9	0033	218	,DC,,0033,
FFDA	0034	219	,DC,,0034,
FFDB	0035	220	,DC,,0035,
FFDC	0036	221	,DC,,0036,
FFDD	0037	222	,DC,,0037,
FFDE	0038	223	,DC,,0038,
FFDF	0039	224	,DC,,0039,
FFE0	0041	225	,DC,,0041,
FFE1	0042	226	,DC,,0042,
FFE2	0043	227	,DC,,0043,
FFE3	0044	228	,DC,,0044,
FFE4	0045	229	,DC,,0045,
FFE5	0046	230	,DC,,0046,
FFE6	0021	231	,DC,,0021,
FFE7	004C	232	,DC,,004C,
FFE8	0052	233	,DC,,0052,
FFE9	0057	234	,DC,,0057,
FFEA	000F	235	TR,DC,,000F,
FFEB	000A	236	TEN,DC,,000A,
FFEC	0009	237	NINE,DC,,0009,
FFED	0003	238	THREE,DC,,0003,
FFEE	0900	239	,SFLG,1,0000,
FFEF	2500	240	,JMPID,,TTYB(1)
FFF0	FF13	241	TTYB,DC,,FF13,
.	.	.	.
.	.	.	.
.	.	.	.
FFFE	2500	242	,JMPID,,TTYB1(1)
FFFF	FF13	243	TTYB1,DC,,FF13,

Appendix E

Definition of Terms

A :	MSP of A	Eqs. (7) and (8)
AL :	LSP of A	Eqs. (7) and (8)
AC0 :	Accumulator 0 of the TDY-52B	
AC1 :	Accumulator 1 of the TDY52-B	
AC2 :	Accumulator 2 of the TDY-52B	
AC3 :	Accumulator 3 of the TDY-52B	
ADPH :	MSP of ADP	Eq. (7)
ADPL :	LSP of ADP	Eq. (7)
ADPH1 :	MSP of ADP1	Eq. (7)
ADPL1 :	LSP of ADP1	Eq. (7)
ADPH2 :	MSP of ADP2	Eq. (7)
ADPL2 :	LSP of ADP2	Eq. (7)
ADYH :	MSP of ADY	Eq. (8)
ADYL :	LSP of ADY	Eq. (8)
ADYH1 :	MSP of ADY1	Eq. (8)
ADYL1 :	LSP of ADY1	Eq. (8)
ADYH2 :	MSP of ADY2	Eq. (8)
ADYL2 :	LSP of ADY2	Eq. (8)
B :	MSP of B	Eqs. (7) and (8)
BL :	LSP of B	Eqs. (7) and (8)
COUT :	Address of clock latch, a store instruction to this address will cause bits 8 through 15 to be loaded in the clock latch (see Figure 4).	

DATA ($2+000F_{16}$) : Used by the flow charts to reference the memory contents at the memory location computed by adding AC2 to $000F_{16}$.

F8 : TDY-52B general purpose output flag 8 which may be set at T2 or reset at T6 under software control. There are 6 flags available.

INTEN : Interrupt Enable Flag when set under software control enables TDY-52B internal interrupt structure.

INTRA : TDY-52B interrupt signal input which is sampled under hardware control during T3 to detect an interrupt. If set the next instruction executed is from memory location 0001_{16}.

K2 : K_2 of Eq. (6)

LIF0 : Refers to the TDY-52B 16 word by 16 bit Last In/First Out Shift register, that is, stack.

LSD : Least Significant Digit

LSP : Least Significant Part

MSD : Most Significant Digit

MSP : Most Significant Part

POUT : Address of Pitch Latch, a store instruction to this address will cause bits 8 through 15 to be loaded in the Pitch Latch (see Figure 4).

ϕ : Gyro Roll position angle.

Ψg : Gyro Yaw position angle.

PSIG : Ψg

PSIG1 : PSIG one sampling period removed.

PSIG2 : PSIG two sampling periods removed, Eq. (8).

PULSE : Refers to the setting and resetting of a TDY-52B Flag during the same microcycle.

REFPH : MSP of current Pitch Reference (see Figure 23).

REFPL : LSP of current Pitch Reference (see Figure 23).

REFPH1 : MSP of starting value of Pitch Reference for line segment ending at time tg_1 (see Figure 25).

REFPL1 : LSP of starting value of Pitch Reference for line segment ending at time tg_1 (see Figure 25).

SEGADD : Address of TG1, TG2 or TG3 (see Figure 23).

SEL : TDY-52B Select Flag.

STACK1 : Reference to location one of the LIF0.

θg : Gyro Pitch position Angle.

THETA : θg

THETA1 : THETA one sampling period removed.

THETA2 : THETA two sampling periods removed, Eq. (7).

TTY : Mnemonic for ASR-33 Teletype.

TF : Time to zero fins.

TF3 : TF + 100 ms, time to initiate lock fins pyrotechnic

TI : Time to initiate actuator pyrotechnic

TI3 : TI+100 ms

TL : Launch time - not used

T2 : Second Stage ignition time

[AC0+AC1] : Refers to 32 bit word formed by AC0, MSP, and AC1, LSP.